HP Service Activator

User's and Administrator's Guide

Edition: V51-1A

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

Legal Notices

Document id: p158-pd019502

Contents

Contents

In This Guide

This guide describes HP Service Activator for users of solutions that are based on the product, operators focused on the service activation processes as well as system administrators. The first three chapters address the needs of process-oriented operators; the rest of the information is primarily for system administrators.

Chapter 1 is a general introduction to HP Service Activator for the audience of this guide. The rest of the manual is focused on the user interface of HP Service Activator. This chapter provides the background to understand the displays shown on the UI and the interactions that can be performed.

Chapter 2 describes the main web browser based user interface of HP Service Activator for operators, allowing to monitor activity in the HP Service Activator system and interact with provisioning processes.

Chapter 3 describes the user interface for the HP Service Activator data repository, known as inventory, which allows you to view and edit the data in the repository and which also has a feature that allows you to start workflows. The feature may or may not be used in your solution.

Chapter 4 describes the Common Network Resource Model as it appears in the inventory views which are included in the product distribution and the data load functions which can be exercised from these views.

Chapter 5 describes the Self Management part of the main user interface, mostly concerned with system householding.

Chapter 6 describes user management.

Chapter 7 describes how the system, mainly the user interface, can be configured.

At the end of the guide are a number of appendices covering a number of details about script files, configuration files, log files, system security and the data collector, a tool to collect information to submit as evidence for support requests.

Audience

This is the only manual in the documentation set for the HP Service Activator core product intended for solution users and system administrators. All the other manuals are intended for system integrators who build solutions based on the core product. If you are interested in that topic it is recommended to begin with *HP Service Activator, System Integrator's Overview*.

Conventions

The following typographical conventions are used in this guide.

Font	What the Font Represents	Example
Italic	Book or manual titles, and manpage names	Refer to *HP Service Activator, Workflows and the Workflow Manager* and the *Javadocs* for more information
	Provides emphasis, introduces a new term	You *must* follow these steps.
	Identifies a variable or parameter	Run the command: `InventoryBuilder` *<sourceFiles>* The *assigned_criteria* parameter returns an ACSE response.
	Location descriptor	*$JBOSS_DEPLOY*
`Computer`	Text and items on the computer screen	The system replies: `Press Enter`
	Command names	Use the `InventoryBuilder` command
	Method names	The `get_all_replies()` method does the following…
	File and directory names	Edit the file `$ACTIVATOR_ETC`/`config/mwfm.xml`
	Window/dialog box names	In the `Test and Track` dialog…
`Computer Bold`	Text that you must type	At the prompt, type: **`ls -l`**
Keycap	Keyboard keys	Press **Return**
`[Button]`	Buttons on the user interface	Click `[Delete]`. Click the `[Apply]` button.
`Menu Items`	A menu name followed by a colon (`:`) means that you select the menu, then the item. When followed by an arrow (`->`), a cascading menu follows.	Select `Locate:Objects->by Comment`

Install Location Descriptors

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

Descriptor	What the Descriptor Represents
$ACTIVATOR_OPT	The base install location of Service Activator. The UNIX® location is `/opt/OV/ServiceActivator` The Windows® location is `<install drive>:\HP\OpenView\ServiceActivator`
$ACTIVATOR_ETC	The install location of specific Service Activator files. The UNIX location is `/etc/opt/OV/ServiceActivator` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\etc`
$ACTIVATOR_VAR	The install location of specific Service Activator files. The UNIX location is `/var/opt/OV/ServiceActivator` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\var`
$ACTIVATOR_BIN	The install location of specific Service Activator files. The UNIX location is `/opt/OV/ServiceActivator/bin` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\bin`

$ACTIVATOR_THIRD_PARTY	The location for new Java™ components such as workflow nodes and modules.
	The UNIX location is `/opt/OV/ServiceActivator/3rd-party`
	The Windows location is
	`<install drive>:\HP\OpenView\Service Activator\3rd-party`
	Customized inventory files are stored in the following location:
	`$ACTIVATOR_THIRD_PARTY/inventory`

$JBOSS_HOME	The install location for JBoss. The UNIX location is `/opt/HP/jboss` The Windows location is `<install drive>:\HP\jboss`
$JBOSS_DEPLOY	The install location of the Service Activator J2EE components. The UNIX location is `/opt/HP/jboss/server/default/deploy` The Windows location is `<install drive>:\HP\jboss\server\default\deploy`
$ACTIVATOR_DB_USER	The database user name you define. Suggestion: `ovactivator`
$ACTIVATOR_SSH_USER	The Secure Shell user name you define. Suggestion: `ovactusr`

1   Introduction

Role of This Manual

The HP Service Activator core product is a framework which provides general capabilities for a range of activation solutions. Not all of the features of the core product are relevant for all solutions.

This manual, being about the core product, stops short of what is specific for your solution. We do not know what services your solution will activate. In several cases there will be statements like "The interaction window can take several forms. Its content is defined as part of the solution." You will therefore need additional documentation for your solution. If the solution is also a product, like the VPN Solution Pack, you will get the additional information from the manuals for that product. If you work with a solution that has been customized for your needs, you will need additional information from the solution delivery project, whether in the form of solution handover training, informal notes or formal manuals.

This manual is for operators who need to monitor and possibly interact with the HP Service Activator system to perform their roles in service provisioning processes and for administrators of the HP Service Activator system. Some of the information is for users in both these categories. Quite a lot of it is only for administrators.

This Introduction chapter briefly explains what HP Service Activator is and does to give you a background for consuming the information in the following more specific chapters. It introduces terminology and concepts that will be assumed to be known.

The Workflow Engine

At runtime HP Service Activator is essentially a workflow engine suitable for service activation. The engine runs provisioning processes which are defined by workflows and focused on activation transactions. A workflow in HP Service Activator is a step-by-step program, where each step is defined by a node in a flowchart. The node performs a basic function, which may be to parse an incoming request message, to retrieve, write or update a data object from the data repository (HP Service Activator inventory), to ask an operator for some information, to perform a calculation to derive a parameter value for an activation, to write an audit trail record and or an informative message to operators, or most importantly, to execute an activation transaction.

An activation transaction will interact with one or more activation targets. Targets are the devices or systems whose states need to be changed in one way or another to activate a service for an end customer. What the targets are will vary from solution to solution. For a solution to activate VPN connectivity services for commercial customers they will be MPLS provider edge and access network devices. For a solution to activate services for residential customers they will be access and aggregation devices and a variety of systems - email servers, voice softswitches, IPTV middleware servers, providing the services.

The HP Service Activator workflow engine has two parts, the workflow manager and the resource manager. The workflow manager knows all workflows that are defined on the system, and executes them. Each execution of a workflow is called a job. Many jobs, also running the same workflow, can run in parallel. When a workflow job reaches an activation transaction the

workflow manager hands it over to the resource manager which performs all communication with activation targets and manages the transaction which may consist of multiple interactions with different targets, each such interactions known as an atomic task. Meanwhile the workflow job will wait for the activation transaction to finish.

Workflow Properties

Each workflow job is a single sequential process. But jobs can start child jobs, the parent and the child job can run in parallel, and there can be multiple child jobs in parallel.

Workflow jobs can run with different priority in the workflow manager, and the priority will be inherented from workflow jobs to activation transactions. Priorities will have an impact in systems with a very heavy processing load. Normally, the workflow engine is not 100% busy; workflow jobs typically spend more time waiting for activations or input than they are busy with internal processing.

Workflow definitions belong to solutions. As part of the definition each workflow has a brief description, and each node within a workflow has a name and also - optionally - a description. A workflow has a number of variables that will be instantiated for each job. There will be solution specific variables, but there are also some predefined ones for general purposes that are common to most workflows. The common variables include an identifier of the order being processed, an identifier of the service (instance) being activated, the type of the workflow (could be name of the solution), the state of the job, i.e. an indication of what the job is presently doing or waiting for. All of this information: step names, descriptions, values of common variables, is extracted from workflow jobs and presented on the user interface to facilitate monitoring by operators of what is happening in the system.

Scheduled Workflow Jobs

Normally, when an order is received to run a workflow, it will start immediately. Workflows can also be scheduled to run once or repeatedly at a later planned time. A scheduled job is different from a running job. It does not contain all the information that is needed for a running job, and so does not occupy resources in the workflow manager in the same way as a running job. On the other hand a scheduled job has a number of attributes which do not apply to a running job or have a different meaning from its similar attributes, including:

- Start Time   Time when the workflow job will be started.

- Repeating Period   If present, specifies that new jobs will be started repeatedly, from the same workflow definition, each time the period expires

- End Repeating   Time when repeated starting of workflow jobs will stop

- Repeating Type   Relative: if one job instance gets started with delay, delay the next one similarly, or Absolute: start time strictly defined by Start Time and Repeating Period

- Start Missed Instance   If system not availble at the time when job should start, then start it with delay when system becomes available again

- Group Id   An identifier used to define a group of scheduled jobs.

- Status   Status defined for the job during its scheduled time.

- Description   Description of the job during its scheduled time.

Inventory

HP Service Activator has its own data repository, also known as inventory, typically divided into resource inventory and service inventory. Resource inventory has data about resources that are activation targets and sometimes must be allocated in a dedicated manner for an individual service

user, for example a DSLAM port. Service inventory records details about services that have been activated, often needed when services are later modified or deactivated/deleted.

Inventory is often intensively accessed by workflows. There is also a user interface that allows operators to browse and edit inventory data, described in chapter 3.

Authentication and User Roles

Visibility of information on the user interface and the ability to perform actions depends on privileges which can be defined for workflows and for inventory data as part of a solution.

The system administrator can define a set of user roles, assign privileges to roles, define users of the system with username and password for authentication on login, and assign roles to users. Each user may have multiple roles, and each role may have multiple privileges. Management of user roles is described in a dedicated chapter (chapter 6).

The system administrator has a predefined role, called 'admin' (the name can be changed by configuration). There is also a predefined role called 'internal'.

The ways in which users can view and control workflows and jobs are controlled by a set of role attributes which are defined in the workflow definition and inherited by each job that executes the workflow. When values are set for these attributes, only users with the specified roles can perform the actions associated with each attribute; if values are not set, any user can perform the actions. There are four role attributes of a workflow/job:

- default role: specifies who can interact with a job; also works as default value for the three following role attributes when they are not specified (which in practice they seldom are)

- start role: specifies who can start a job directly from the UI

- trace role: specifies who can view jobs in Active Jobs view

- kill role: specifies who can stop jobs by action from the UI

The identity of a user is authenticated by checking username and password at login on the user interface. The user's roles are established at this time. Authentication is performed by a module of the workflow manager. It is possible to configure which module will be used. HP Service Activator has a native authentication module (class `DatabaseAdvancedAuthModule`). It is also possible at installation time to configure Service Activator to support Single Sign On when the UI is cross launched from other applications, i.e. to accept the user's identity as established on the launching system. If supported by the other application Single Sign On will also apply when cross launching is done from Service Activator.

A Typical Workflow

To give you an idea of a typical workflow, here is a description of a common pattern, depicted in Figure 1-1. The pattern may actually be implemented by means of several workflows, one which performs the northbound interaction to receive and respond to an order from an external system, one or more others dedicated to specific activation work.

In an actual workflow, each box as shown in Figure 1-1, will require several workflow nodes, the number depending on the complexity of the details of the process. At the high level, as shown, the process is typically as follows:

- A customer order is entered into a CRM system, where it is validated, approved and forwarded to the next level of processing. In a simple case it can go directly to Service Activation. In a more complex case, an order management process may be needed to decompose an order which is for a bundle of services and to separate the request for automatic activation, which can be handled by HP Service Activator, from other tasks which must be performed manually or passed on to partners of the provider.

Figure 1-1   **Typical Service Activation Workflow**
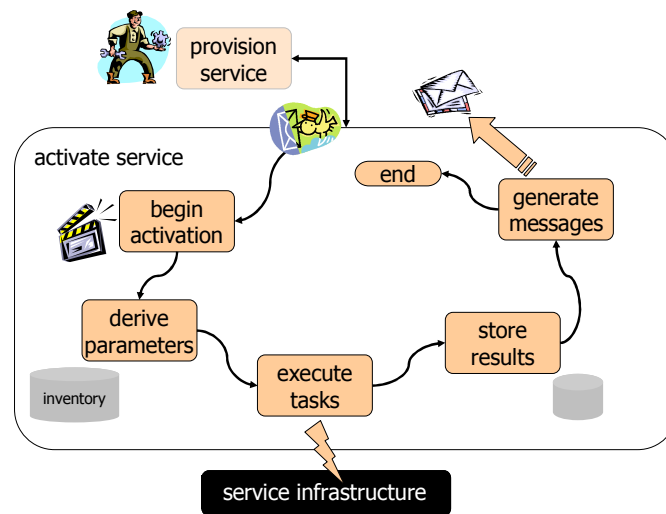
- The activation system based on HP Service Activator receives the request for activation of a service, maps to the appropriate workflow and starts an instance of the workflow as a job.

- The workflow job inspects the parameter values from the request and calculates or gathers additional necessary parameter values.

  — For example, if a service type is specified as 'Gold' or 'Silver', the corresponding technical values for bandwidth and similar parameters will be looked up. For a complex technology, the derivatioin can also be complex.

  — Resources needed to satisfy the request, such as an access port, for example on a DSLAM, where customer premises equipment can be connected, are allocated from resoource inventory. Resource identifiers will also be activation parameters. Allocated resources are recorded in resource inventory.

  — Parameter values, for example selection of a device or port, can also be obtained from a dialog with an operator.

- When all parameters are ready, the necessary activations are executed, ideally as a single task, which may be compound.

  — If an atomic task within a compound task fails, any preceeding atomic tasks are automatically rolled back. This logic is not visible in the workflow.

- If the activation was successful, the newly activated service will be recorded in service inventory. If it failed, any reserved resources are again released.

- A summary of the action of the workflow, including service id, main parameter values, success or failure, is recorded in HP Service Activator's audit trail.

- A response message to be returned to the requester is prepared. Success or failure indication and any relevant parameters that were derived by the workflow are included in the message. The message is forwarded according to the protocol of the northbound interface. This may be done by a sender module or by the web service interface after the workflow terminates.

- The response message is processed by the order handling system or CRM system. This is outside the scope of the activation system.

2   Operator User Interface

HP Service Activator's user interface is web browser based. It does not use any software components that must be installed on the client system prior to use.

The user interface is configurable in many respects as described in chapter 7 and may therefore appear different from what is described in this chapter.

HP Service Activator Main UI Window

After logging in (see next section, "Login"), you will see HP Service Activator's main UI window as shown in Figure 2-1.

Figure 2-1   **HP Service Activator UI Main Window**
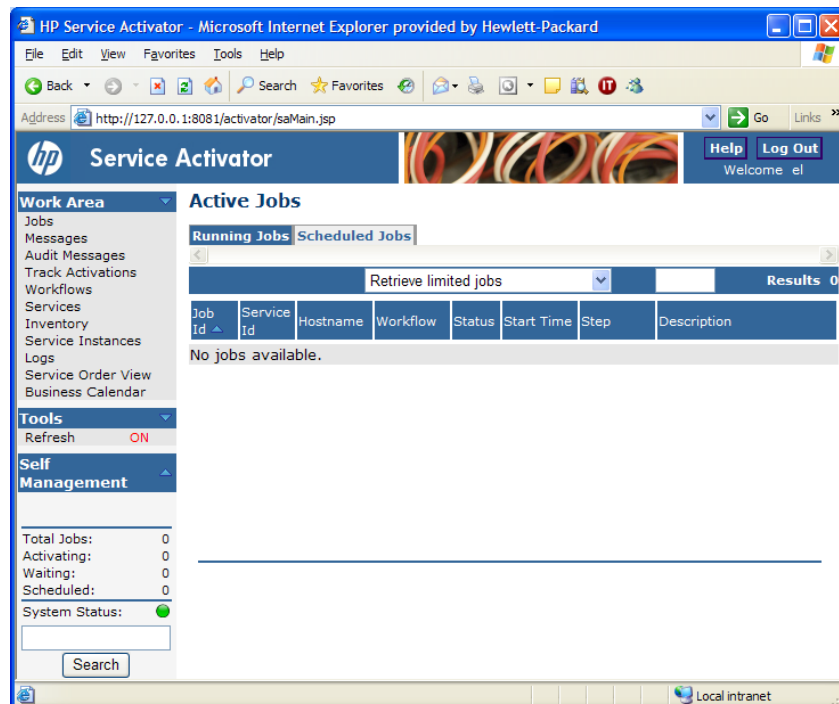
In addition to the main window some menu items will launch additional windows for specific purposes, including the inventory UI window which is described in chapter 3. Other launchable windows will be described in this chapter.

The main window has three areas:

- The header bar at the top welcoming the user who is logged in.

- The navigation pane on the left which shows a menu of available functions and summary statistics. The menu is configurable and may deviate from what is shown in Figure 2-1.

- The main working area whose contents is selected from the menu.

Most menu selections will populate the main working area with the selected view, which may be Jobs view, Messages view, etc. When this happens, the previous contents of the area will be overwritten. Some of the views, like Inventory view, will appear in separate windows which are launched if not already open when the selection is made.

The views that can be selected from the Work Area menu generally display lists of information. The Jobs view display lists of jobs which may be active or scheduled, etc. Some of the lists allow you to define column value filters to select the information that is relevant to you. Typically the lists are organized under different tabs. In some cases there are right-click menus associated with each row in a list, allowing you to perform actions on the object that is shown in the row, for example to interact with a running job, or edit an entity in inventory.

What you can do from the user interface: the menu items that are available to you, the items shown in lists in views you select, the actions you can perform from right-click menus, all may be controlled by the roles that have been assigned to you.

NOTE   If you do not see menu items, jobs, or queues that you expect to see—or if you are unable to interact with a specific job or workflow—check if the appropriate role has been assigned to you and that you are logged in as the appropriate user.

More information about the header bar and the navigation pane are given in the following subsections.

Header Bar

The header area contains the [Help] button, the [Log Out] button, and a welcome message. Pressing [Help] pops up a small menu with two options: About and User's Guide. The About option brings up information identifying your version of Service Activator. The User's Guide option opens this guide in an Adobe Acrobat Reader (available for free download at www.adobe.com).

The [Log Out] button terminates your session with HP Service Activator and releases resources used by the session.

A login session with Service Activator will terminate by timeout if there is no activity in the window for a sufficiently long period of time, by default set to 15 minutes. Before timeout, a warning will be presented in the header area.

Navigation Pane

The navigation pane in the left part of the main UI has three menus, an area with job counters, a system status indicator, and the Search function.

You can collapse or expand each of the menus by clicking the small triangle shown to the right of the menu name.

Work Area Menu

The Work Area menu appears in the Operator UI for all users. The contents of the menu can be configured, and some items can be restricted based on user roles. The default menu contains these items:

- **Jobs**   shows Active Jobs view in the main working area

- **Messages**   shows Messages view in the main working area

- **Audit Messages**   shows Audit Records view in the main working area

- **Track Activation** shows Track Activation view in the main working area

- **Workflows**   shows Workflows view in the main working area

- **Services**   shows Services view in the main working area

- **Inventory**   shows Inventory UI in a separate window

- **Service Instances**   show Service Instances view in a separate window

- **Logs**   shows Logs view in the main working area

- **Service Order View** shows Service Order view in a separate window

- **Business Calendar**   shows Business Calendar in main working area

Most of the views that are shown in the main working area contain mostly information presented in list form: lists of workflow jobs, messages, audit records, etc. Each row in a list will present information about one entity (job, workflow, etc.) with a number of attributes organized in columns under a column header. In many cases you can perform actions on the entities represented by the rows. Right-click on the row and a menu of action options will appear, then click one of the options. You can also generally sort the lists by the values appearing in any column. To trigger the sorting, click on the column header. If you double-click you will invert the sorting order.

All the views selectable from the Work Area Menu, with the exception of the Inventory UI which is described in chapter 3, are described in subsequent sections within this chapter.

Tools Menu

The Tools menu always appears in the Operator UI. By default has only one item, **Refresh**, which toggles screen refreshing on and off. It applies to the information shown in lists in the main working area. Most of these lists are refreshed at same rate which is configurable and defaults to 45 seconds.

Self Management Menu

For users without administrator privileges, this menu will contain only one item, Change Password, which allows you to change your password. The menu and the functions that a system administrator can select from it are described in chapter 4.

NOTE	Change Password is only available if HP Service Activator's native authentication module is being used.

Job Counters

In the bottom left corner, you can find the job counters:

- **Total Jobs** indicates the number of running jobs.

- **Activations Ongoing** indicates the number of jobs engaged in an activation transaction.

- **Awaiting User Input** indicates the number of jobs waiting for user interaction.

- **Scheduled Jobs** indicates the number of jobs scheduled to start in the future.

The refresh rate of the job counters is configurable separately from the refresh rate for information shown in views. It defaults to 4 seconds and cannot be disabled.

System Status

The system status "traffic light" shows at a glance by color the "health" of the HP Service Activator system, which may be distributed over multiple hosts, called nodes. Here is the color code:

- green   running, in a cluster one or more nodes may have been locked or suspended by the administrator

- blue   all nodes have been locked or suspended by the administrator

- yellow   one or more nodes out of order

- red   all nodes out of order

- grey   unable to retrieve status from nodes

To the left of the "traffic light" you may see an Info (as shown below) or Warning icon. This icon signifies that one of the system database tables for Messages, Audit Records or Database Messages has grown past a configured threshold (see chapter 7) and needs to be cleaned up.

Search

The search function allows you to search the contents of the main window for a text string. Enter a string in the box, and click the [Search] button. If the text is found, the first instance will be highlighted. If you click the [Search] button again, the next instance will be highlighted, and so on.

Login

To log in to HP Service Activator from your browser you must direct the browser to the URL:

```
http://<HPSA-host>:<HPSA-port>/activator
```

where <HPSA-host> is the name or IP address of the server where JBoss with HP Service Activator is hosted, and <HPSA-port> is the port number for the JBoss web server. By default it is 8080, but a different number may have been specified at installation time.

NOTE   With HP Service Activator running on a Windows server, if you want to access the UI from a browser on the same platform, you can launch the browser and direct it to HP Service Activator by selecting
`start -> All Programs -> HP Service Activator -> Service Activator`

When your browser is directed to HP Service Activator, the login screen appears first, as shown in Figure 2-2.

Figure 2-2   Service Activator Login Window
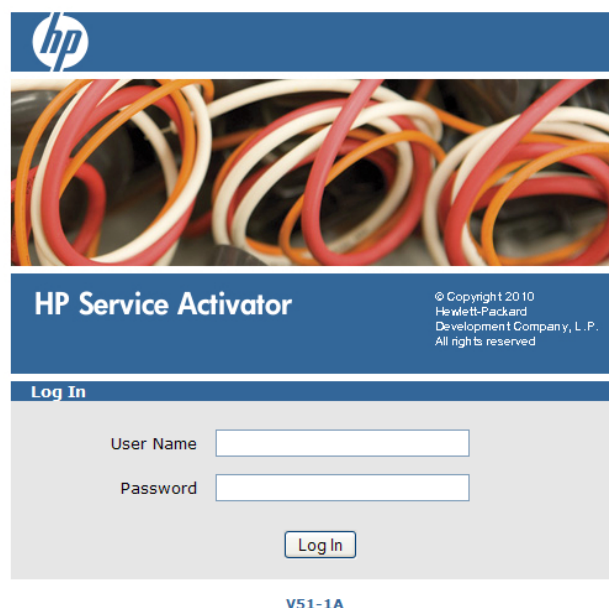
NOTE   If authentication of users has not been enabled (this is the case immediately after installation of HP Service Activator) any username will be accepted, and it is not necessary to type a password.

NOTE   The system administrator may set up a new user so that the password must be changed on first login. The workflow manager can be configured to expire user password after a specified number of days, in which case users will also be alerted to change their passwords.

When a valid user name and password have been entered and the [Log In] button pressed, the main window of the UI will appear as shown in Figure 2-1.

NOTE   Failure to log in because HP Service Activator is not running properly on the server, for example if the Oracle database cannot be accessed, will appear as failure to authenticate the username & password.

User Interface Integration

An installation of HP Service Activator can be integrated with other systems so that UI views of one system can be cross launched from the UI of another system. For example, graphical network views or tabular device views of NNMi or device configuration views of NA can be launched in context in the Inventory View of HP Service Activator. Examples are discussed in chapter 4. You can also launch HP Service Activator views from NMMi, or other systems if configured as part of a solution.

HP Service Activator is integrated into HP's Light Weight Single Sign On framework. This feature must be configured at installation time. When it is active you can cross launch UI views without being presented with a login screen for the system you are launching to. The user identity from the system you first logged in to will be transferred to and accepted on the target system. User identities must be configured on all cooperating systems and the same user names must be used on all systems.

If Single Sign On is not configured, cross launch is still possible, but the first time within each session you open an HP Service Activator window by cross launching, you will be presented with the login screen as discussed in the previous section, and you must enter user name and password for a valid HP Service Activator user.

When cross launching into the UI of HP Service Activator, you can bypass view selection in the navigation pane and launch directly into one of the views which are normally selected from the work area menu, for example the inventory view or the service order view. The launching system must be configured to know the appropriate URL; that is typically accomplished as part of system integration using information found in *SI manual*.

Active Jobs View

Upen login to HP Service Activator, the Active Jobs view is shown in the main working area, and you can select it again at any time by selecting `Jobs` in the Work Area menu. The Active Jobs view shows the jobs that are known in the workflow engine. You can only see jobs for which you have the trace role.

A job is an execution of a workflow. Jobs can be running or scheduled. A scheduled job has been created but not started. It does not occupy resources in the workflow engine like a running job.

A running job can be waiting for an activation transaction to complete or for a synchronization event, i.e. data to be input from an operator through an interaction with the job or from an external source, or completion of a child job.

The synchronization point where jobs wait for events are called queues. Queues have names which are defined for the solution. A couple of queue names are built-in to the core product: a queue named sync is normally used by a workflow which is waiting for a child job to terminate; a queue named activation is used by jobs waiting for an activation transaction to complete. The jobs list is organized by tabs showing queue names. When a queue tab is selected, only the jobs waiting on that queue are listed. In addition to queue names, the following tabs are predefined:

Running Jobs   all active jobs, i.e. jobs that have been started and have not finished, are shown under this tab

Scheduled Jobs   jobs which have been created in the scheduled state, but not yet started

NOTE   The scheduler module of the workflow manager can be disabled in the configuration of the workflow manager. In that case, the Scheduled Jobs tab will not appear.

The number displayed on a queue tab is the number of jobs waiting on the queue. By default, a limited number of jobs are displayed in the list. However, in the drop-down menu on the bar immediately below the tabs you can alternatively select 'Retrieve specified number of jobs' and enter the number you want or select 'Retrieve all jobs'.

View Queued Jobs

In the Active Jobs view as shown in Figure 2-3 (compare to Figure 2-1 to see how the list has been cut out from the complete browser window) there are four tabs, Intro, sync, Running Jobs and Scheduled Jobs. The Intro queue tab is selected, it refers to a queue used by a workflow to wait for a user interaction. There is also a popup menu, it is described below under "Right-Click Actions for Queued Jobs". The following description applies generally to the situation where a queue tab is selected.

Figure 2-3   **Active Jobs View - Queued Jobs**

NOTE	The blue and purple icon in fron of the Job id only appears for those jobs that require user interaction.

The jobs list contains a number of values for each job, shown in several columns. The set of columns and their order are configurable. The default columns for a list of queued jobs, as shown in, Figure 2-3 are as follows:

- Job Id   Unique number assigned to the job when it is created
- Service Id   Id of the service being activated
- Hostname   Name of the server which is executing the job
- Workflow   Name of the workflow being executed
- Status   Status of job execution, one of:
  Running, exceuting a workflow node
  Waiting, waiting for an activation or queue interaction
  Transitioning, moving to next workflow node
  Finishing, ending the workflow.
- Start Time   The time the job started
- Post Time   The time from which the job has been waiting on the queue
- Step   The name of the node where the workflow is waiting
- NodeDescription   Description of the node from the workflow definition (it is often empty, as in the screenshot in Figure 2-3)

If configured, the following columns can also be shown:

- Order Id   Id of the order being processed
- Type   Type of the workflow
- State   State of the job

NOTE	Your roles may restrict which jobs are visible in the Active Jobs view. You can only see a job on a queue tab if you have the default role for the job, allowing you to interact with it. To see a job on the Running Jobs tab you must have its trace role (which may be equal to the default role).
	There is a special queue tab for jobs which are waiting for activation transactions to complete, labelled 'activation'. This tab is normally invisible, because the role required to interact with a job during an activation is 'internal', and it is not recommended that operators should have the internal role. However, this role assignment may be overridden in the workflow definition in order to make the job visible on the activation queue tab.

### Right-Click Actions for Queued Jobs

You can right-click on one of the jobs in the list to perform an action on the job. A popup menu with five options will appear, as shown in Figure 2-3.

NOTE	The activation queue has an extra action on the right-click menu, View Activation. It launches the Transaction Details window, see Figure 2-13, with details of the activation transaction that the job is waiting for.

Interact with Job. You must have the default role for a job to interact with it. When you interact with a job, an interaction window appears. The details of the interaction window will depend on the solution. It contains a title; a row describing the job with headers (blue) and data columns (grey) similar to the jobs list; optionally an information panel, possibly with multiple tabs; optionally some rows where you can enter or edit values for variables, optionally some radio button rows allowing you to select one of several options for the next steps of the workflow logic.

An example including an information panel, but no variables or radio buttons, is shown in Figure 2-4; in this case the workflow job needs to wait for the operator to perform an action before it can proceed with automated activation.

Figure 2-4 **Job Interaction Window**
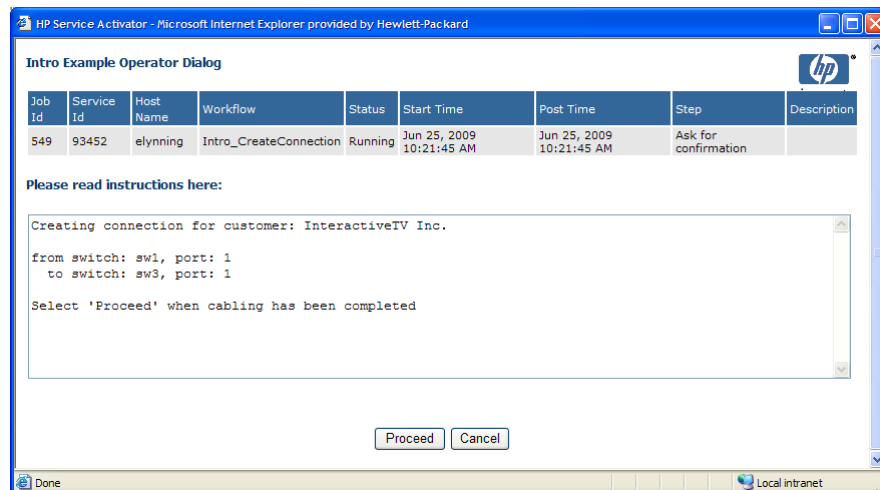
NOTE   Do not interact with a job on the activation queue.

Stop Job. If you have the kill role for the job, you can stop it.

Stop Job (Forced). If you have the kill role for the job and adminsitrator privileges, you can issue the Stop Job (Forced) command. Use this option only when the job you are trying to stop does not respond to the Stop Job command. The Stop Job (Forced) action tries to stop the job normally. If the job does not stop normally within 1 second, it is forced to stop.

Change Roles. An administrator can change the role required to interact with a queued job, labelled 'Request role' in the popup window, see Figure 2-5. Its definition may come from the workflow node which controls the interaction (the node named in the Step column of the Jobs list); this role will restrict which users can select 'Interact with Job'. You can change it to make an exception for the interaction the job is waiting for. It is also possible to change the default role of the job (not just for the interaction it is waiting for).

Figure 2-5 **Change Roles for Queued Job**
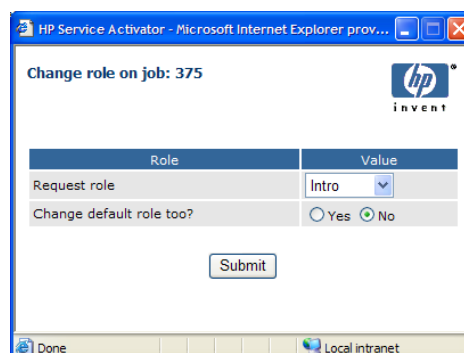
Change Priority. This action, which is only available if you have administrator privileges, pops up a window where you can change the priority of the job. The higher the number, the higher the priority.

View Running Jobs

If instead of a queue tab you click on the Running Jobs tab in the Active Jobs view, all running jobs that are visible to you (limited by trace role) will be listed (possibly limited by the number of jobs if there are many), as shown in Figure 2-6.

Figure 2-6   **Active Jobs View - Running Jobs**

The columns shown in the jobs list are almost the same as when you selected a queue tab. There is no Post Time, and Node Description is replaced with Description, which is now the description associated with the entire workflow, not a step where it is waiting.

Right-Click Actions for Running Jobs

When you right-click on a job in the list of running jobs a slightly different set of actions appear in the menu than the list shown in Figure 2-3 for a job from a list on a queue tab. The actions for a job selected under the Running Jobs tab are Stop Job, Change Roles, Stop Job (Forced) and Change Priority. Only Change Roles is different from the actions described for queued jobs.

Change Roles. If you have administrator privileges, this action pops up a window, as shown in Figure 2-7, which allows you to change the role attributes of a running job. The role attributes are originally specified as part of the workflow definition.

Figure 2-7   **Change Roles for Running Job**

View Scheduled Jobs

If you click on the Scheduled Jobs tab in the Active Jobs view, all scheduled jobs will be listed, as shown in Figure 2-8.

Figure 2-8   **Active Jobs View - Scheduled Jobs**

Active Jobs

Running Jobs | Scheduled Jobs

| Retrieve limited jobs | | | | | | | Results 1 - 2 |

Job Id ▲	Service Id	Workflow	Start Time	Repeating Period	End Repeating	Group Id	Status	Description
396	117	Intro_CreateConnection	Jun 16, 2009 1:00:00 PM			100	Good	Create connection at 13:00
397	117	Intro_DeleteConnection	Jun 16, 2009 1:10:00 PM			100	Good	Delete after 10 mins

The information in the list of scheduled jobs is different from the information for queued or running jobs. A number of special attributes which exist only or have special meaning for scheduled jobs are included: Start Time, Repeating Period, End Repeating, Group Id, Status, Description. See the desciption of scheduled jobs in chapter 1 for more information.

Right-Click Actions for Scheduled Jobs

The right-click menu for a scheduled job includes the following action options: Start Job, Modify Job, Delete Job.

Start Job changes the start time to start a scheduled job immediately.

Modify Job opens a pop-up window, similar to the one in which you can edit job scheduling attributes; see Figure 2-15. The Job Id and the Workflow fields are not editable.

Delete Job deletes a scheduled job.

Messages View

You can select the Messages view by selecting `Messages` in the Work Area menu of the navigation pane.

Workflow jobs can post messages for reading by operators in the Messages view. Each message is placed on a named tab. The tabs are also known as queues, but should not be confused with queues where workflow jobs can wait for interactions or activations as discussed under "Active Jobs View" above. The tab names and the contents of messages depend on the solution. Different tabs are used to separate messages about different topics. When no messages have been present on a tab for some time, the tab will not appear.

Normally messages are used to provi de information to operators about errors or exceptions that have occurred in provisioning processes, but it is possible from a workflow to put a message with any content.

Figure 2-9 shows an example of the Messages View with two messages on a tab named Intro.

Figure 2-9　　　　**Messages View**

Job Id	Service Id	Order Id	Hostname	Workflow	Post Time ▼	Type	State	Step	Message
563	93452	1076	elynning	Intro_Controller	Jun 26, 2009 10:44:18 AM	Intro	Post_actvn	Print message to operator	Message sent to CRM: <response_msg> <header> <message_id>1076</message_id> </header> <body> <code>ERROR</code> <status>1</status> <description>Resource failure</description> </body> </response_msg>
564	93452	1076	elynning	Intro_CreateConnection	Jun 26, 2009 10:43:57 AM	Intro	Reserving	Notify operator 1	No free ports found for connection between switch sw1 and switch sw3.

If there are many messages the list will be divided into multiple pages. The page size is configurable. You can use the three buttons on the top bar ([<Prev],[Next>], [Go]), to browse back and forth among the pages or go to a specific message.

The list has a number of columns. The information in each column is as follows:

- Job Id　　　　　Identifies the job that posted the message.

- Service Id　　　Id of the service being activated

- Order Id　　　　Id of the order being processed

- Hostname　　　Name of the server which is executing the job

- Workflow　　　The name of the workflow the job was executing

- Post Time　　　The time the message was posted by the job

- Type　　　　　Type of the workflow

- State　　　　　State of the job

- Step　　　　　The name of the node in the workflow that posted the message.

- Message　　　The actual message that was posted, controlled by the workflow.

If you have administrator privileges, you can use actions in a right-click menu to delete a single message or all the messages in a queue.

NOTE　　　　　Delete messages when action has been taken and the messages are no longer needed to avoid having to find relevant messages among large numbers of historical ones.

Audit Records View

You can select the Audit Records view by selecting `Audit Messages` in the Work Area menu of the navigation pane.

HP Service Activator's audit trail consists of records written to document events that have taken place on the system. It constitutes the authoritative history of what the system has done over time. Which records are written and the detailed information they contain depend on the solution. Recorded events can include:

- users logging into and out of the system

- inventory data created, modified or deleted

- actions completed by workflow jobs

The latter category is considered the most important one, since the main work of an activation solution will be done by workflow jobs. Workflows can be configured so that audit events are recorded when a workflow job starts or ends, with no detailed information. The more interesting workflow events are explicitly generated within the workflow and may include a number of fields providing information about the task accomplished by the job such as order id, service id, customer, and parameters detailing the service.

Audit records are stored in HP Service Activator's database until they are deleted by an administrator, who can also export them to an external system; see chapter 4. As long as they are in HP Service Activator's database you can view them in the Audit Records view. Select Audit Messages in the Work Area menu of the main UI, and the audit reord list will appear. An example is shown in Figure 2-10. Each row represents one audit event record.

Figure 2-10   **Audit Records View**

Audit Messages

| | | | Reset | Filter | << | < Prev | 1 - 6 / 6 | Next > | >> | | Go |

Job Id ▼	Service Id	Order Id	Type	Date	Event Type
				Jun 26, 2009 10:58:31 AM	USER_LOGIN_EVENT
				Jun 26, 2009 10:55:46 AM	USER_LOGOUT_EVENT
	sw1||4			Jun 26, 2009 10:55:28 AM	INVENTORY_EVENT
562	93452	1076	Intro	Jun 26, 2009 10:43:34 AM	CONNECTION_CREATED
560	93452	1076	Intro	Jun 26, 2009 10:42:18 AM	CONNECTION_CREATED
558	93452	1076	Intro	Jun 26, 2009 10:41:53 AM	CONNECTION_CREATED

The audit record list has several columns which contain values of attributes of the audit events. It can be configured to contain additional columns with further attributes that are not shown in Figure 2-10. The attributes shown here are:

- Job Id   Unique id of the workflow job that caused the event, when applicable

- Service Id   Identifier of the service being activated (service id variable of the job). In inventory event: primary key of the inventory entity

- Order Id   Identifier of the order being processed

- Type   Type of workflow being processed

- Date   Time the event was generated

- Event Type   Type of the event. The following types are predefined: USER_LOGIN_EVENT, USER_LOGOUT_EVENT, INVENTORY_EVENT, KILL_JOB_EVENT, KILL_JOB_NULL_USER_EVENT, START_JOB_EVENT, END_JOB_EVENT, EXCEPTIONAL_END_JOB_EVENT,

START_JOB_NULL_USER_EVENT.
Events generated explicitly by workflows have solution-specific event types; an example is CONNECTION_CREATED in the screenshot above.

The additional attributes that can be configured but are not present in Figure 2-10 are:

- Workflow State   State of job when event was generated

- Hostname   Name of the server where the event originated

- Workflow   The name of the workflow the job was executing

- Step   The name of the node in the workflow that posted the event

- Message   Solution-specific, defined in workflow

- Class   For inventory event, Java bean class of the data entity

- Username   For events originating from the user interface: login name of the user

Browsing and Filtering

The number of of audit records found in the system can be large. The bar at the top of the list has seven buttons ([Reset], [Filter], [<<First], [<Prev], [Next>, [>>Last], [Go]), which allow you to filter audit records or navigate through multiple pages. By default, 60 audit records are shown per page; the number is configurable. The [Filter] button launches a window where you can specify a search filter in terms of values that must match attributes of the audit records you want to see. Enter as many attribute values as you need to narrow your search. After you have entered the filter values, click the [Submit] button. When you use a filter, the filter attribues values are displayed at the bottom of the audit record list. Click the [Reset] button to turn off filtering and go back to full list view.

Audit Record Details

The information shown in the list is a summary of each audit record. A window showing the full details of the audit record pops up if you double-click on the record in the list. This window will include any attributes that are not shown in the list as well as additional variables and associated values that may be present in a record generated by a workflow. An example is shown in Figure 2-11.

Figure 2-11   Audit Record Details Window
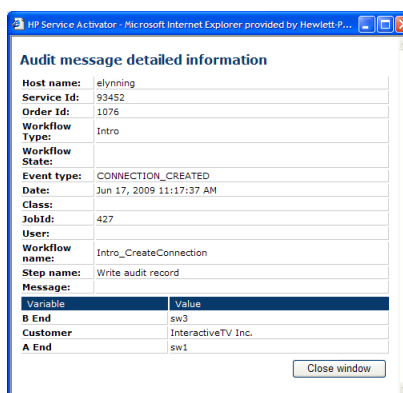
You can also open the audit record details window by right-clicking an audit record row in the list. In the pop-op menu select Open instance or Open new instance to view the details. Selecting Open new instance opens each record in a new separate window; Open instance will overwrite the initial window if you open additional records.

Track Activations View

You can select the Track Activations view by selecting `Track Activations` in the Work Area menu of the navigation pane.

An essential phase in a workflow job occurs when it executes an activation transaction, i.e. an activation task comprising one or more so-called atomic tasks, each of which is an interaction with an activation target, such as a network device or service platform. Activation transactions typically have a longer duration than other steps in a workflow, up to several minutes when many commands are executed against a device or the target system is executing a time-consuming request. You may therefore want to be able to track what is happening during an activation transaction. This is what the Activation Tracking view helps you to do. An example of the view with one active transaction is shown in Figure 2-12. There is one row per currently active transaction. Each job can only have a single active transaction at a time, so when there are multiple rows in the list, the Job Id value will be different in each one. You can right-click on a row and select `View Activation Details` to view more detail about the progress of a single transaction. A separate window pops up with this information as shown in Figure 2-13.

Figure 2-12   Track Activation View

Track Activations

Job Id ▼	Service Id	Order Id	Workflow	Start Time	Priority	Atomic Task	Lock Status	Lock Arguments	Hostname	Transaction State
23	site001	mess001	VE_AddSite	Jun 30, 2010 3:50:03 PM	0	GLOBAL.DemoRouter_Connect_Site	Acquired	[GLOBAL.DemoRouter, 10.1.1.1]	elynning	ACTIVE

The information shown in each column of the activation transactions list is as follows:

- Job Id   Identifier of the workflow job that is executing the transaction
- Service Id   Identifier of the service being activated
- Order Id   Id of the order being processed
- Workflow   Name of the workflow the job is executing
- Start Time   The time the transaction started
- Priority   Type of workflow being processed (from type variable of the job)
- Atomic Task   The atomic task the transaction is currently executing. A transaction may go through multiple atomic tasks.
- Lock Status   Acquired or Waiting. For each atomic task the transaction may need to acquire a lock within the relevant plug-in. When one transaction holds the lock, another one may be waiting for it (depending on the number of transactions allowed to share the lock).
- Lock Arguments   Transactions which access a plug-in with the same locking arguments compete for the same lock.
- Hostname   Name of the server the job and transaction are running on.
- Transaction State State of the transaction.

Type and State of the workflow job can also be shown, if configured.

Figure 2-13 **Transaction Details**

The Transaction Details window repeats some of the information from the row in the transaction list and adds Global Transaction Id, a transaction identifier which is unique across multiple hosts in a cluster, and the atomic task level activity detail shown as coloured rows at the bottom, each one starting with 'DO'. Each of these rows completely specifies an atomic task by its name and parameters. The row color indicates state of each atomic task. A task in progress appears in grey, tasks completed successfully appear in green, tasks which encounter errors and must roll back appear in orange.

Workflows View

You can select the Workflows view by selecting `Workflows` in the Work Area menu of the navigation pane.

The Workflows window lists those workflows that are deployed on the HP Service Activator system for which the user has the start role. It is possible to start workflow jobs from this list, but that feature is generally not used for an operational solution. It may be convenient for testing and debugging. Normally a workflow job will require some variables to be initialized, for example with values received in a message from a requestor system, before the job is started, and that will not be done when a job is started from the workflow list; hence it will not run properly.

Workflows in the list may appear in three different background colors. Grey color marks correct workflows, yellow color marks workflows which were loaded by the workflow manager with warnings but are otherwise correct (for example, two workflows with the same name exist), red color marks workflows which contain errors. If your workflow was loaded with warnings or errors, the problem causing those warnings or errors is described next to the workflow in the Description column. Otherwise the Description text is taken from the workflow definition.

Right-Click Actions on Workflow

By right-clicking on a workflow in the list you get a popup menu of actions you can perform on a workflow. Figure 2-14 shows an example workflow list and the action menu. The actions are described in the following.

Figure 2-14   **Actions in the Workflows Window**

Start Workflow. You can start the workflow as a job, but you cannot initialize its variables.

Schedule Workflow. You can create a scheduled job to run the workflow. This action opens the Schedule job form (shown in Figure 2-15), where you can enter values for the attributes of the scheduled job (described in chapter 1).

Figure 2-15 **Schedule Workflow Job Window**

Reload Workflows. If you have administrator privileges, you can force the Workflow Manager to reload all deployed workflows. This will update the workflow list.

Services View

You can select the Services view by selecting `Services` in the Work Area menu of the navigation pane.

This view focuses on the service affected by running workflow jobs, taking off from the service id variable values whereby each workflow job can identify the service it is activating. The view simply shows a list of the different values for this variable which are in use in active workflow jobs. You can right-click on one of the values and select to view a list of either running or scheduled workflow jobs whose service id variable has that value.

Service Instances View

You can select the Service Instances view by selecting `Service Instances` in the Work Area menu of the navigation pane.

A solution may use simple service instance records to store information about services it has activated. Each service instance record is identified by a service identifier whose value would typically come from the service id variable of the workflow job that activated it. In addition it will contain values of other variables from the workflow.

Services instances can be shown in the service instances view, which opens as a separate window from the main UI, as shown in Figure 2-17. If any service instance records exist, a plus-sign will appear to the left of the 'Service Instances' line under 'Contents'. Click, the plus-sign, and a list of service identifiers for all existing service instance records will open.

To view details of a service instance, click its identifier. You will note that only string-valued variables can be properly displayed. You can delete a service instance by clicking the associated Delete icon.

Figure 2-16   **Service Instances Window**
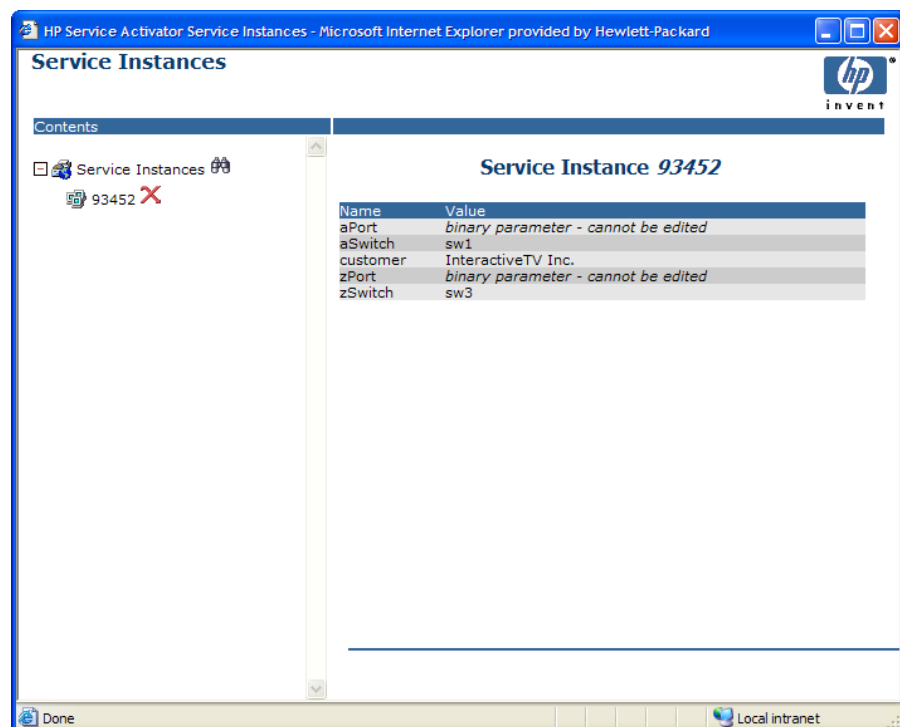
Logs View

You can select the Logs view by selecting `Logs` in the Work Area menu of the navigation pane.

The two components of HP Service Activator's workflow engine, the workflow manager and the resource manager, produce extensive log files as they run. A maximum size is configured for the log files. As the files reach the maximum size, they are closed and renamed with a number in the name. Then a new file is opened with 'active' in the name. Thus the numbered files represent the history and can be cleaned away if there is no need for them.

The log files are not significant for normal operation; they are intended for troubleshooting. Log files are written as XML documents. The format is described in Appendix C. A viewer that presents log file contents in readable form is available in the Logs view. An example of the view is shown in Figure 2-17. The names of the active and historical log files which exist - for the category selected by the tab at the top - are shown at the bottom of the view. Hover the mouse over one of the file names, and the last modification date of the file will be shown as a tip.

If your system consists of a cluster of server hosts, beware that separate log files are written on each server host. Select the host you want from the drop-down list at the top of the view.

Then begin by selecting the type of log you want to see from the tabs at the top of the view. Select `MWFM` for workflow manager or `RESMGR` for resource manager. Then at the bottom select the file you want to view, most often it will be the active one. A scrollable read-only list of log entries will then appear.

Note the Filter drop-down box. Depending on how you system has been configured it may be possible to select different filters that will be applied to the log entries in the selected file.

Red log entries represent fatal errors; yellow log entries represent warnings. Grey log entries represent normal conditions.

NOTE   There are no right-click actions available in the Logs view.

Figure 2-17   Logs Window

Service Order View

The Service Order view combines functions of the Active Jobs, Messages, Audit Records and Track Activations views, and lets you focus on a specific order, service instance or type of workflow. The view is shown in a separate window that is launched when you select `Service Order View` in the Work Area menu of the navigation pane. It lets you define a filter based on job id, order id, service id, workflow type and workflow state. You can enter values for one or more of these items, with the exception that job id cannot be used in combinations. Then press `[Query]`, and jobs that match the filter will retrieved and shown.

The jobs list works in the same way as in the Active Jobs view the same tabs and the same action menus. Figure 2-18 shows an example of the Service Order view with a Jobs list where the `add_site` queue tab has been selected.

Figure 2-18 **Service Order View, Queue Tab**

Next to the Jobs tabs there are Messages and Audit tabs you can select to retrieve lists of messages and audit records, respectively, that match your filter. Message lists are a little different from the Messages view in that all messages passing the filter are shown in a single list; the names that would have been on the tabs are shown in a `Queue` column.

Figure 2-19 shows an example with a list of audit records filtered on the workflow type VPNEx., i.e. all of these audit records were written by workflow jobs of that type.

Figure 2-19 **Service Order View, Audit Records**
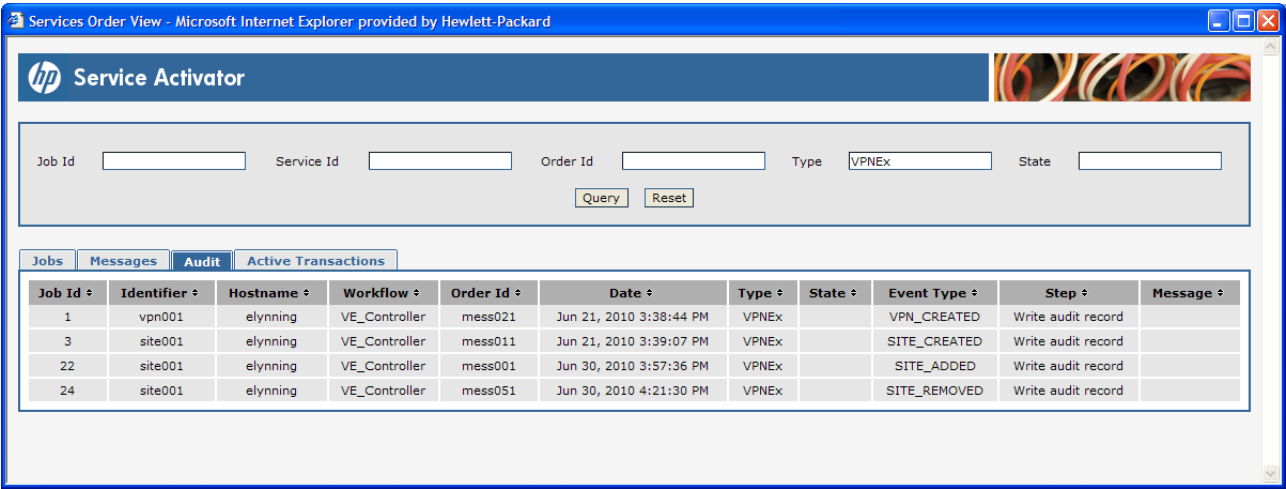
Business Calendar View

HP Service Activator workflows may take your organization's business calendar into consideration and avoid activity, in general or of a specific kind, for example user interactions, outside of business hours on working days. It is optional for a solution to make use of the business calendar.

You can select `Business Calendar` in the Work Area menu of the navigation pane to launch an editor for the days and hours of the business calendar in a separate window. The window appears as shown in Figure 2-20.

Figure 2-20   Business Calendar View

It is possible to define and distinguish between several business calendars. You select in the Operation pane at the top of the window whether to edit an existing calendar, which you then select from the drop-down list, or create a new one, for which you must enter a name. A workflows will refer to a calendar by the name given to the calendar when it is created. You can also select the time-zone in terms of which you will define the daily hours of the calendars.

The Calendar pane has two panes, for weekly schedule (shown in Figure 2-20, partly filled in) and holidays, respectively. In the weekly schedule you must define for each day of the week either start and end times for the workday, or that the day is a weekly holiday. On the holiday tab you can define annual holidays in addition to the weekly ones. You can define holidays as recurring year-by-year or for a single year only.

NOTE   After specifying the holidays for a given year, click `Submit` to save them before changing the year to specify holidays for the next year.

3　Inventory User Interface

The special window that can be opened to access inventory data allows you to browse, search and view the data held in Service Activator's solution data repository (inventory) and to perform operations related to the entities that are modelled in the inventory.

The view for browsing entities is based on an explorer-style tree. Each entity is represented by a branch; branches can be expanded to show child branches and collapsed again. Trees can also include branches which do not represent data entities, but only serve to provide an intuitive and navigable tree structure.

The Service Activator core product does not contain any data entity definitions nor any specific tree definitions. These items must be prepared for each solution by the solution customizer, or they are provided in solution packages.

Based on analysis of the customer's environment and requirements the customizer must design a suitable set of data entities and define one or more trees for the presentation of the data. The definition of the tree structure will take into account the relationships between the entities that are modelled.

Web Service Provider Example

In the following description of the features of the user interface we use a small example with six entity classes that illustrate some real world aspects that can be modelled with inventory data, and how entity relationships are used to structure the presentation tree.

The example is drawn from the world of a web service provider. The six classes are:

- **Room:** represents a room where operation equipment is kept.

- **Rack:** represents an equipment rack for routers and servers; the rack has a limited capacity, so space is reserved for the units of equipment that are located in it, and it has a reference to the room it is placed in.

- **Vendor:** has name and other information about a vendor of switch equipment.

- **Switch:** represents an Ethernet switch and has information such as IP address and password needed to enter commands to control the device.

- **Unix WebServer:** represents a UNIX computer used as Web server and has information such as IP address and port where the Web service is accessed.

- **Win WebServer:** a Windows-based Web server.

Tree Structures and Tree Views

A tree structure is designed to present the inventory data. It uses the structure shown in Figure 3-1. Each branch in the diagram, except the leaves Switch, WebServer and Vendor, represents a potential parent branch with child branches in the actual entity tree. A branch marked with (+) represents an entity class. The branches that are not followed by (+) are title branches which only

serve to provide structure to the tree that is presented. You can see that the structure uses the following parent-child relationships between entities: room-rack, rack-switch and rack-web server.

Figure 3-1   **Structure of Example Tree**
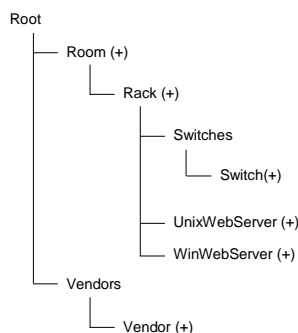
Two views of the trees can be seen in the Inventory UI window:

- **the instance view**, where all the actual data entities are shown as instances of the branches representing entity classes,

- **the class view**, where the title branches are eliminated and exactly when branch is shown for each branch that represents an entity class.

The instance view can be a huge tree if there are many data entities, often too large for all the branches to seen together. The class view will be small. The structure tree in Figure 3-2 can be reduced to the class view by removing the tile branches 'Switches' and 'Vendors'. We keep a root branch, otherwise we would get two disjoint trees.

The Inventory UI window is divided in four rectangular frames, quadrants. Class views of trees are always shown in the upper left quadrant and entity views in the lower left quadrant. When an empty window has been opened, you begin by selecting a view of a tree in the view menu which appears when you point the cursor to either the word 'Class Views' or 'Instance Views' at the top left. After opening both the class view and the instance view of the example tree and then expanding a few branches in the instance view you may see the window shown in Figure 3-2.

Figure 3-2   **Class and Instance Views**
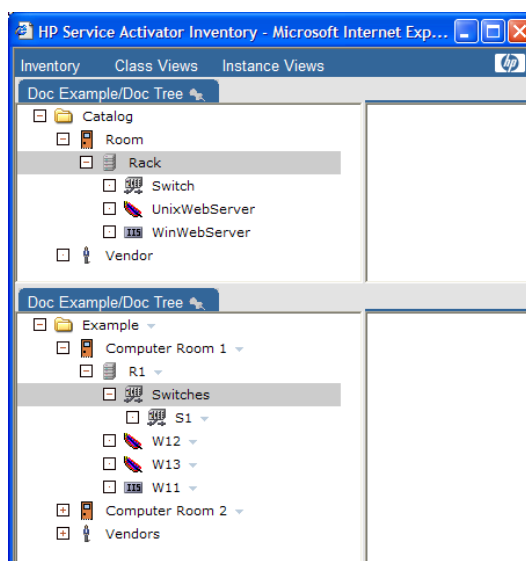
Searching for Instances

Now you can use the class view as the starting point for searches to find specific data instances that may not be easy to locate just by opening branches in the instance view. You can make two kinds of searches: simple and advanced. Locate the class of the instance you want in the class view, right click on it, and select either simple or advanced search in the menu that appears. In general, which kinds of search can be performed on each class in the tree depends on the solution.

Simple Search

If you select a simple search on class Switch, a form where attribute values of the desired instance can be entered appears in the upper right quadrant as shown in Figure 3-3.

Figure 3-3   **Simple Search**

If you enter 'Apache' for the serverType attribute and click Search a list of matching instances is retrieved as shown in Figure 3-4, with the attribute values of each instance shown in a row.

Below the search results you can see that they can be exported to files in different formats. Just click on the name of one of the formats to activate the export function.

Figure 3-4   **Results from Simple Search**

Now you can select one of the search results, each one being an entity instance. Just click on the row. This will bring up in the instance view area (lower left quadrant) that part of the entire tree which has the selected instance as its root. You can see the result also in Figure 3-4. Suppose you selected an instance with a huge number of siblings in the tree, then it would be impractical to view it together with its parent and the siblings. In this way you can get a manageable section of the tree to work with. In the example here the selected instance has no children, so as root it is left all alone in the tree.

Advanced Search

Venturing next to try an advanced search for a Switch, you get the search form shown in Figure 3-5. This form allows you to build a search by specifying match values for the attributes not only of the target entity, but of any of its ancestors as well, according to the class tree. Class, attribute and exactness of equality (=, != or 'like') with the value you specify are selectable in the form. For example, to find all the switches in Rack R1 (in Room 1), select Rack for Bean, name for Attribute, and '=' for Exact. Then enter 'R1' as the Value and click Add to add this condition to your search. You could add more conditions, but for this example that is not needed, so click Search to execute. Figure 3-5 shows the specification of the search (Current search) together with the list Using the Operator User Interface results to a file or select one of them to study its descendants and perform operations from the instance view tree.

Figure 3-5   **Advanced Search**

Saved Search

An advanced search may be cumbersone to enter, so you may want to save it to be able to rerun it later. For this, click Save and you will be allowed to enter a name and a description for the search function you have defined in a popup window as shown in Figure 3-6.

Figure 3-6   **Save advanced search**
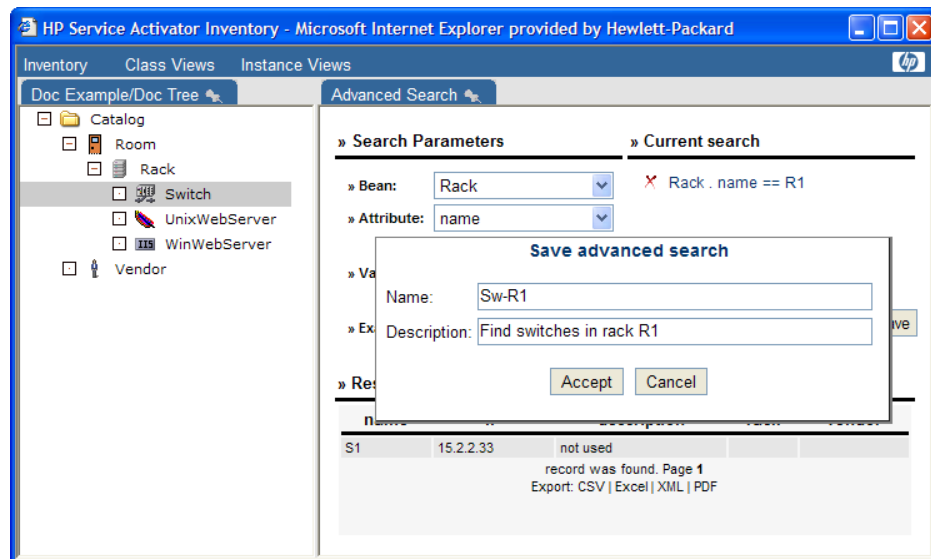
When you have saved one or more searches, or an administrator has made them available to you, an additional menu Stored Searches will appear at the top of the window, next to Class Views and Instance Views, as shown in Figure 3-7. It will have submenus for different trees, such as the Doc Tree shown here. The submenus will contain the actual searches, such as Sw-R1 shown in Figure 3-7. Click on the name of a search, and it will be executed against the current contents of the inventory, as when we clicked the Search button above.

Figure 3-7   **Stored Search**
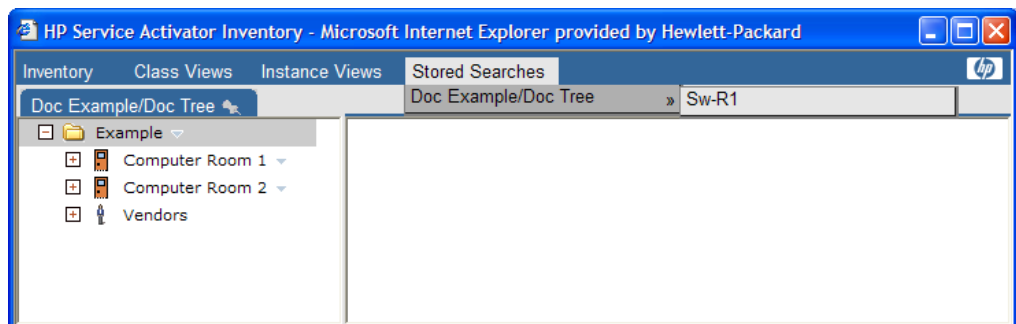
Filtering Instances

The definition of a tree may make use of the filter feature. A filter consists of a number of variables. The values of the variables can be used in definitions of branches to determine the instances which will appear in the tree. Normally the effect of the filter is to establish criteria that will cause some branches to not be shown so that you can more easily focus on the remaining branches.

Filter variables can be set in a pop-up filter editor, as shown in Figure 3-8. The filter editor can be called up by right-clicking in an empty area of the instance tree and selecting Set Filter in the small menu that appears. The filter can be applied from the pop-up window (click Apply). This will typically cause a number of branches to be hidden from the tree. The filter shown in Figure 3-8 has only a single variable, called vendor. Its value can be selected to equal the name of any one of the

vendors defined in the data model. Filter variables are organized in groups. Here there is only one group, the default group.

Figure 3-8   **Filter Editor**
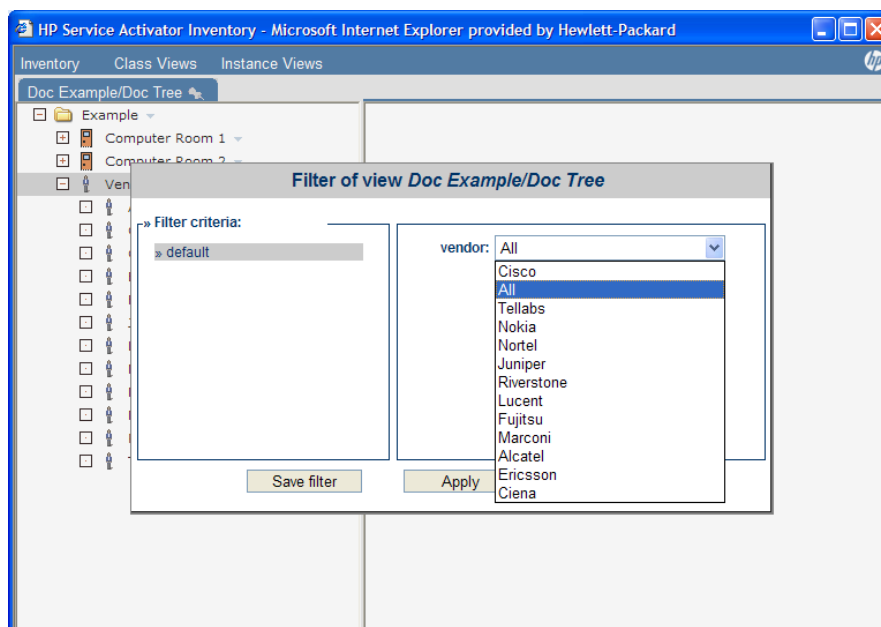
A filter instance, i.e. a set of values for the variables, can also be saved with a name. This allows the filter to be reapplied by selecting it as an item in the Stored Filters menu of the inventory window, as shown in Figure 3-9, where a filter stored with name V-Cisco is being selected. The system administrator can make a filter stored by one user available to other users, or even apply it for them as a mandatory filter.

Figure 3-9   **Stored Filters Menu**
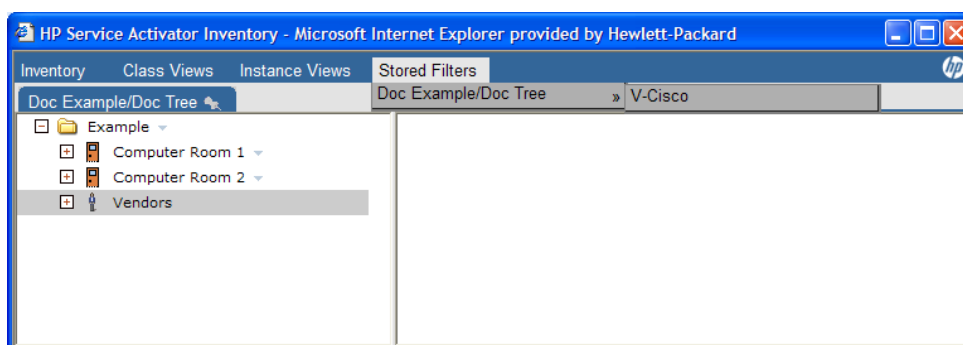
Instance Operations

Instance operations are launched by right-click menus on instance branches in the instance view tree (lower left quadrant) and use forms appearing in the lower right quadrant. The standard operations are Create, View, Edit and Delete. For reservable resources, reserve and release may also be possible. Exactly which operations are available on each branch is defined by the solution.

Branches may come and go in the instance tree as a result of operations you perform to create or delete inventory objects. In such cases the instance tree will change automatically. Changes may

also happen in the background, as a result of workflows running or the interactions of other users. The tree will then not be automatically updated. You can refresh a tree, i.e. synchronize it with the contents of the inventory repository in the database, by clicking its tab above the open tree.

View Operation

For some branches a default operation is defined which can be launched just by clicking on the branch. For example, View Switch can be launched in this way.The resulting form for Switch S1 is shown in Figure 3-10. Note that the upper quadrants have been collapsed here. As the word says, the View form can be viewed; there is nothing further to activate from this form. It shows the names, values and descriptions of the attributes of the class, in three columns. The names and descriptions are the same for all instances of the entity class; both are defined by the solution, the description being optional, so it can be left out when the name says it all.

Figure 3-10   **View Operation Form**

Create Operation

Before you can view an entity it must be created. This can happen in various ways, by loading data into the inventory from an external source or by an action of a workflow. Depending on solution, it can also be done by an operation from the inventory UI. Typically the Create operation appears on the right-click menu of a branch which will be the parent or other ancestor of the new instance branch. In the example tree, the Create operation for a switch appears on the Rack branch. For example, you could launch it from the R1 instance branch to get the Create operation form shown in Figure 3-11.

Figure 3-11   **Create Operation Form**

You create the new instance by entering values for the attributes and then clicking OK. For some attributes (here rack and vendor) where possible values can be known a priori, either defined by the system integrator, or because of an entity relationship, you can choose the value from a drop-down list. The rack is preselected as R1, because the operation was launched from the R1 entity branch. The preselected vendor value here is arbitrary. Mandatory attributes for which a value must be entered are marked by an asterisk.

Edit Operation

Once you have created an entity, you can view it, as we have seen, you can also edit it. The Edit operation will - except possibly in some unusual cases - be found in the right-click menu of the instance branch. Launch it, on Switch S1, and the Edit operation form will appear in the operation area as shown in Figure 3-12. It shows the current attribute values and allows you to edit them before submitting your changes by clicking [OK]. Greyed out values are shown for information but cannot be changed.

Figure 3-12   **Edit Operation Form**

Delete Operation

The last standard operation is Delete. When you launch a Delete operation the attribute values of the instance will be shown in the operations form, and you will be asked to confirm. Depending on the solution, deleting an instance may cause cascade deletion of additional instances related to the first one.

Start Workflow Job

In addition to the standard operation which work only on the inventory, it is possible to customize for a solution operations that allow you to enter parameters and start a workflow job. The workflow can perform any actions. The parameters provide initial values for variables of the job. Parameter entry fields may be dropdown or type-in. They may be prepopulated and editable or empty. Parameter fields can also be read-only. An example is shown in Figure 3-13.

Figure 3-13   **Start Job Operation Form**

When you click [Ok] a workflow job will be started. Depending on the solution the operation may or may not wait for the job to finish. If it does not wait, the job id of the started job will be displayed, allowing you to monitor the job using the Active Jobs or Service Order view. When the operation waits, a progress symbol will be shown until it completes, and then results extracted from the finished job will be shown.

Using Tabs

It is possible in each of the four quadrants to have several tabbed trees or forms open simultaneously. To use this feature you must click the pin icon on the tab to prevent the visible tree or form from disappearing when you open another one. The pin will change to an X when you do this. When you click the X, the tree or form is deleted.

Extended Attributes

In general, the attributes (fields) that are stored in the database and shown on the various operation forms for each entity class are defined by customization. For some entity classes, there may be a requirement for the user to be able to add attributes to the class. Such user defined attributes will not be known to workflows, so they will not have any impact on activation processes, but they can store and display information that the user wants to associate with an instance. When a class has the ability to add user defined attributes, the operation to add them can be invoked from the class view tree and will use a form that is shown in the upper right quadrant (same quadrant as searches) as shown in Figure 3-14 for the Room example class.

Figure 3-14   Adding User Defined Attributes to a Class
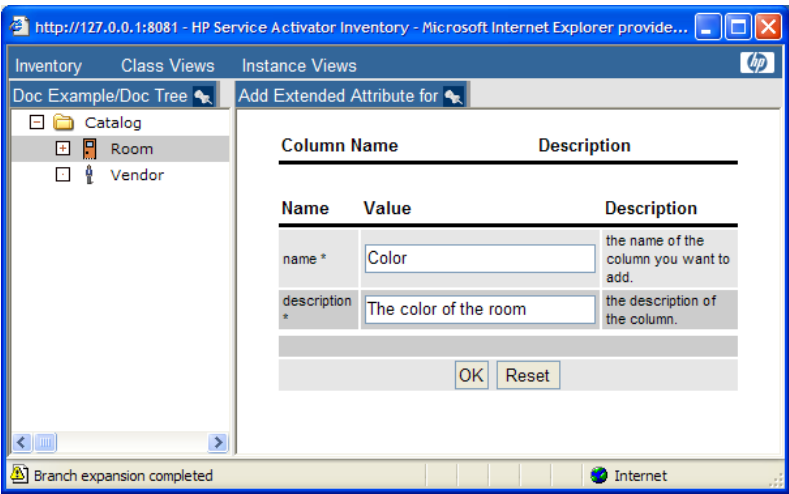
Once you have added your own attributes to an entity class they will appear in Edit and View operation forms. An example is shown in Figure 3-15.

Figure 3-15   Using User Defined Attributes
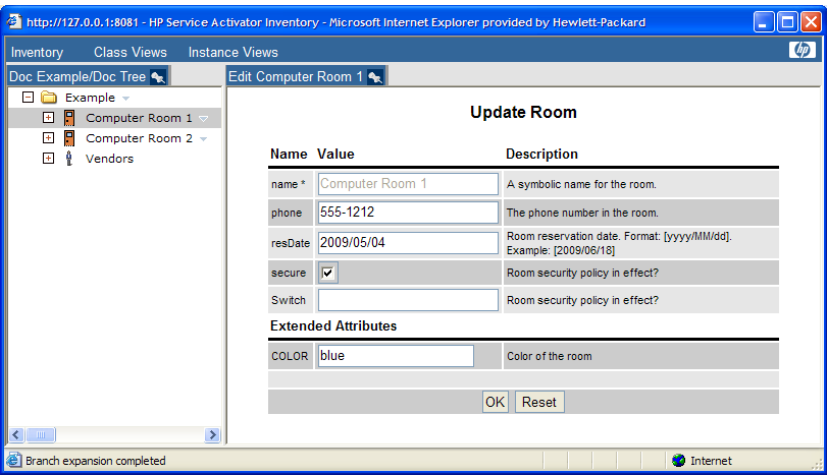
Scrolling in Long Lists of Sibling Entities

When one branch in the instance view tree has a very large number of child branches it is convenient to view the list of child branches using a scrolling area, where only a limited number of branches are shown at a time. An example of a scrolling list of instance branches is shown in Figure 3-16.

Figure 3-16   **Scrolling Branch List**
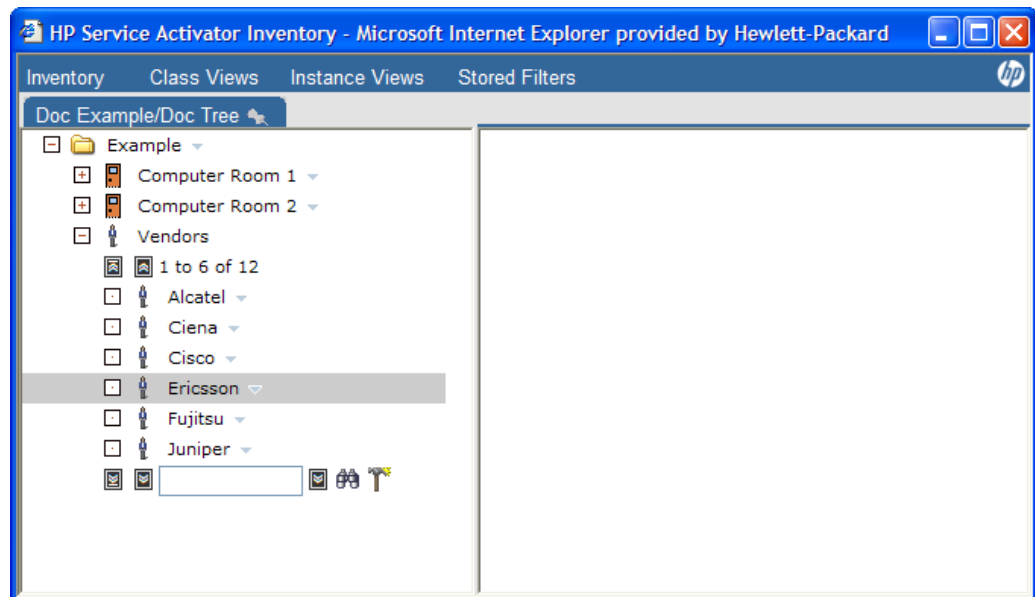
By clicking on the icons to the left above and below the instance branches you can go to the first / last or previous / next group of branches. By typing a value in the field under the branches and then clicking one of the icons to the right of the field you can:

- scroll to a numbered instance (numbering starts from 1),

- scroll to the first instance whose label contains the string you entered,

- eliminate (filter out / flatten) all instances whose labels do not contain the string you entered.

4 Inventory Views of Common Network Resource Model

The Common Network Resource Model (CNRM) is a predefined data model that may be used, possibly in modified form, by a solution. It is visible on the inventory UI in the form of three trees. This chapter describes the object classes of the model along with the three trees:

- Equipment tree   This is the primary tree; it contains all the proper network resource entities. Special operations to upload interfaces directly from network elements and to create aggregate interfaces are launched from this tree.

- Parameter tree   This tree contains entities which are used to configure the CNRM itself, the connection of HP Service Activator to NA and NNMi, and the NNMi dataload function; the details of the tree and the aspects which can be configured are described in the section "Parameters Tree" below.

- NNMiDataload tree   This tree is used to control the NNMi dataload process; see the section "Dataload from NNMi" below.

The data model is not described here with all data classes down to individual fields. Refer to *HP Service Activator, System Integrator's Overview* for more detail.

Equipment Tree

The main classes of data shown in the equipment tree are networks, network elements, interfaces on network elements, and links between network elements. The root of the tree breaks into regions, defined as administrative units by the service provider. Regions and locations where network elements can be located are defined in the parameter tree; regions cannot be defined in the equipment tree.

The example equipment tree shown in Figure 4-1 has just one region, called Denmark. In that region there is one network, called Lab. The network elements in a network fall under two branches according to the value of their Role field: 'PE Routers' and 'Other Network Elements'. In the Lab network there is one network element in each of these categories.

The inventory object (resource) representing a network element has a number of fields recording properties of the device. As an example the network element named c3600-1.dnk.hp.com is shown in Figure 4-1. All of the information represented here was obtained by the dataload process, either coming from NNM, added by mapping or enrichment, or as default values for new objects.

Figure 4-1   **CNRM Equipment Tree with Network Element**

Most branches in the Equipment tree which represent objects have the standard View, Edit and Delete operations, applying to the object the branch represents. As shown in Figure 4-2 the network element branches also have operations which crosslaunch views of NNM and NA.

Figure 4-2 **CNRM Equipment Tree with Network Element Operations Menu**
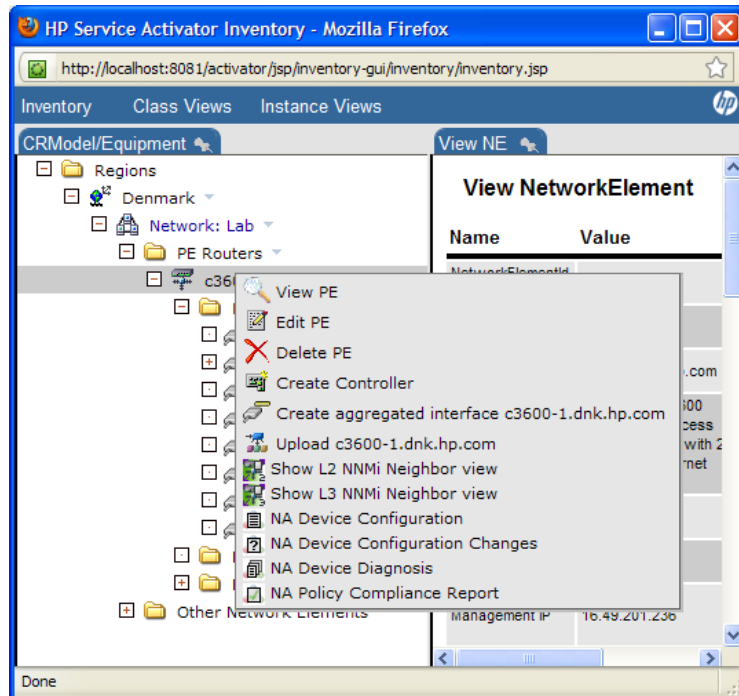
Depending on the browser and the operations, these operations will launch the indicated view into the operation forms quadrant of the inventory UI window or in a new pop-up window. You may be presented with a login screen for the launched application to enter username and password, but only the first time during a session. With NNMi this can be avoided if both the HP Service Activator system and NNMi have been configured with single sign-on. As an example, Figure 4-3 shows the NNMi L3 neighbor view of a network element.

Figure 4-3 **CNRM Equipment Tree with NNM View**
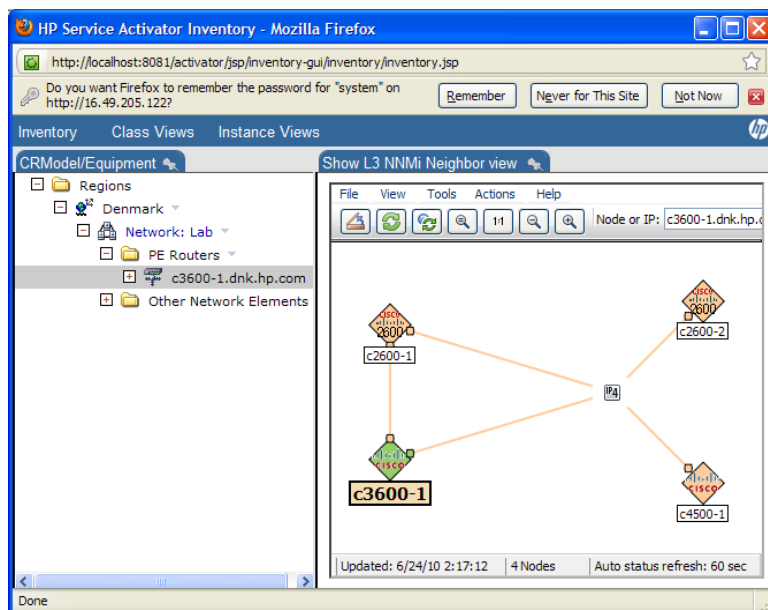
Interface Upload

The interface upload operation is performed by direct interaction betweem HP Service Activator and network elements. It serves to create relevant interfaces which exist on a network element, but are not yet modelled in the CNRM. Launch this operation from the operation menu on the network element branch as shown in Figure 4-2, enter a slot number in the form which appears, if you want to upload interfaces only from a specific card slot, and click ok to launch the workflow which will query the device, parse the response and create any interfaces which were not modelled already.

Create Aggregate Interface

The special operation to create an aggregate interface on a network element is launched from the operation menu on the network element branch as shown in Figure 4-2. A form as shown in Figure 4-4 then appears where you can select the type of interface to aggregate and add individual interfaces to the aggregate from available ones. The name of the aggregate interface will be generated according to device convention (shown here as 'Port-channel1'). When you click ok, a workflow will run to execute the commands to create the aggregate interface as specified.

Figure 4-4   **CNRM Equipment Tree Create Aggregated Interface**
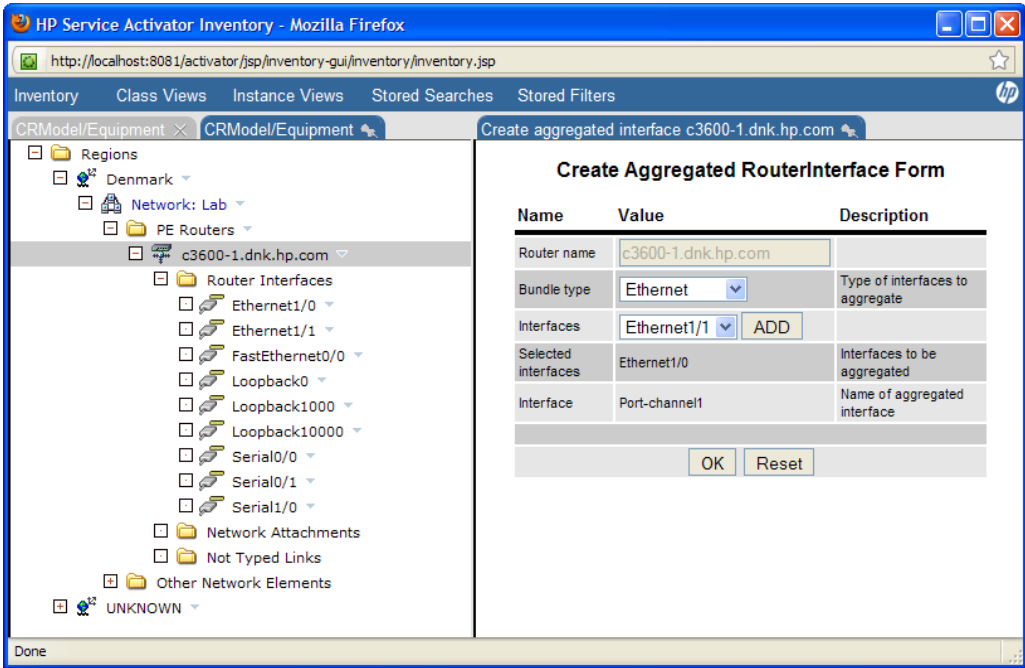
Interface Branches

Further in the breakdown structure of the tree we see for each network element its interfaces, attached access networks and links. The first level of interfaces shown is physical interfaces (ports), with logical sub-interfaces such as individual VLAN-tagged ones shown at the next level. Interfaces and sub-interfaces are represented by the same data model object class and use the same View, Edit and Delete forms. The details for a VLAN sub-interface are shown in Figure 4-5.

Figure 4-5   **CNRM Equipment Tree with Sub-Interface**

Link Branches

A link is a trunk connection between two network elements. In general a link can carry traffic for many service connections. In the tree a link can be shown in the context of both of the network elements that it terminates on. The fields of the link object indicate the network elements and termination points (interfaces) at its endpoints. Figure 4-6 shows an example.

Figure 4-6   **CNRM Equipment Tree with Link**
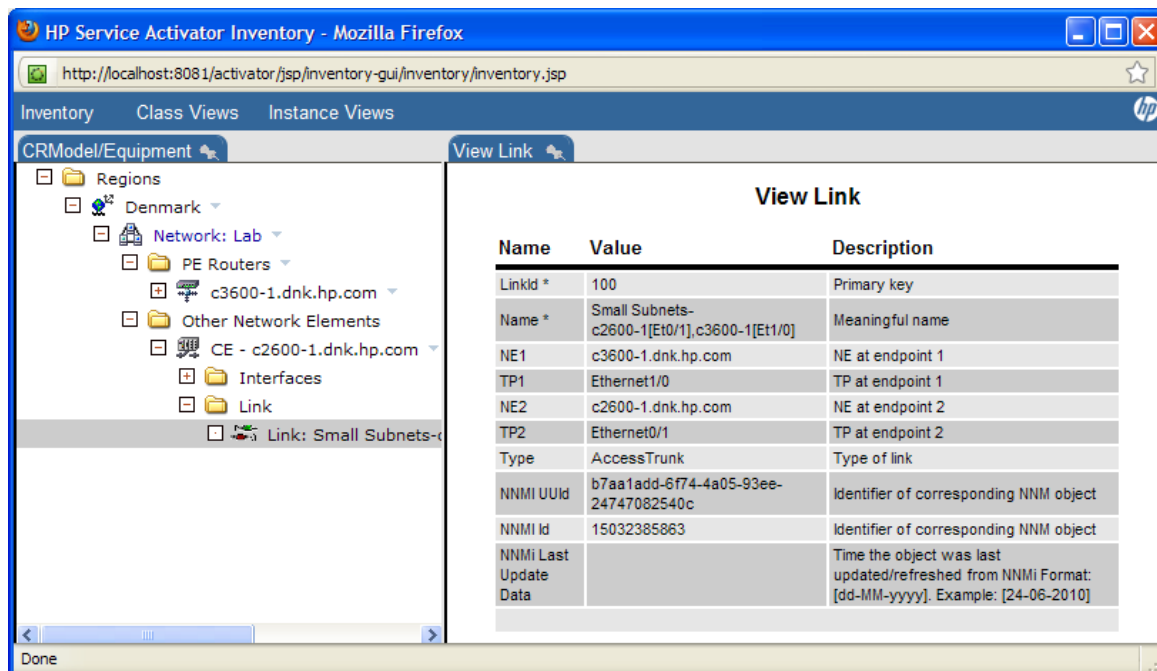
Dataload from NNMi

The NNMi dataload function can be used to maintain the contents of the CNRM based on an associated installation of the NNMi product. CNRM data is loaded from and synchronized with the NNMi data model. There is a dedicated tree, the NNMiDataload tree (shown in Figure 4-7) to control this function. It is is launched by selecting the 'Run Dataload Workflow' operation on the root branch of the tree and then accepting (click ok) to run the workflow.

The dataload function supports two scenarios:

- initial load starting from an empty model

- incremental load of entities which have been added to the NNMi network model since previous load

There is another function, launchable by the operation 'Run Dataload Check Nodes Workflow' from the root node of the NNMiDataload tree. This workflow will run through the network elements in the CNRM and check that each one still exists on NNMi. If not, the AdminState of the network element is set to "Down".

The dataload function retrieves information network elements within a defined scope and loads information which is not yet present in the CNRM. In the second scenario some entities may be loaded from NNMi with attributes which differ from those already present in the CNRM. The user must then decide whether to load the new attribute values or keep the existing ones.

The function has two phases: an automated data retrieval phase and a manual reconciliation phase.

In the automated phase data entities are first retrieved from NNMi, mapped to be compliant with the CNRM and placed into intermediate tables. Then the data is enriched with additional field values according to configured rules, adding information which is not obtained from NNMi. Intermediate entities which are exactly equal to existing CNRM entities are then dropped; intermediate entities which do not match any existing CNRM entities are created as new entities in the CNRM, and entities which already exist, but with differing field values are displayed in the NNMiDataload tree for user decision.

In the manual phase the user must inspect the entities in the NNMiDataload tree, possibly modify some field values, and accept all desired entities. Changes can be rejected per network element. To accept some changes, but not all, on subobjects of a network element, first accept some of them individually, then reject the rest.

In the NNMiDataload tree (see Figure 4-7) the Dataload Status shows if a dataload process (workflows) is ongoing, and what phase it is in. You will need to click on the tab for the tree to refresh the shown status. When the status is Reconciliation the user must inspect and take action on all the objects which are shown as leaves under the network branches (known networks as well as Unknown Network for network elements which have not been assigned to a network).

The form in the right hand side shows the comparison window for fields where the value from Dataload (DL) differs from the one stored in CNRM (CR). The interesting fields are marked with pink. Consider the Description field. Here the value loaded from NNMi (from the configuration of an interface on a network element) begins with "** HPSA VPN **". In the CNRM this value has been changed (by manual editing) to "---". Now you can choose to keep the edited value in the model by pressing the '<<' button, which will cause the value on the right (CR) to overwrite the one on the right (DL), and then clicking OK at the bottom of the form to modify the data object in the intermediate (DL) model. Whether or not you make any changes you must select the Accept operation to store the intermediate data object in the actual CNRM.

Figure 4-7   NNMiDataload Tree

Parameters Tree

In this tree you can configure several aspects of the CNRM and of the integration of HP Service Activator with NNMi and NA.

Administrative Units

In this subtree you can define regions and locations. Each network object must be created in a unique region. Locations must also be defined to be within regions, but a location does not need to belong to a unique network.

Equipment Types

In this subtrees are defined the vendors, OS versions, and element types which can be valid values of the respective fields on the network element object. The list of vendors is prepopulated with all equipment vendors known to NNMi. If you find the long list unmanageable, you can delete some vendors you are sure you will not be using.

Under each vendor you will find, or you must define, the relevant OS versions and element types, both of them arranged in groups, where a group includes a number of OS versions or element types which are equivalent for management purposes.

Interface Types

Similar to equipment types, all interface types which are valid values of the interface type field on the Interface object are defined by the values shown in this subtree. The prepopulated set of values will normally be sufficient.

NA and NNM Integration

This subtree has two signleton branches, containing parameters that must be set for HP Service Activator to communicate with NA and NNMi, respectively.

The NA parameters are shown in Figure 4-8; they are explained by the comments in the Description column of the view/edit form.

Figure 4-8   **Parameters Tree NA Configuration**
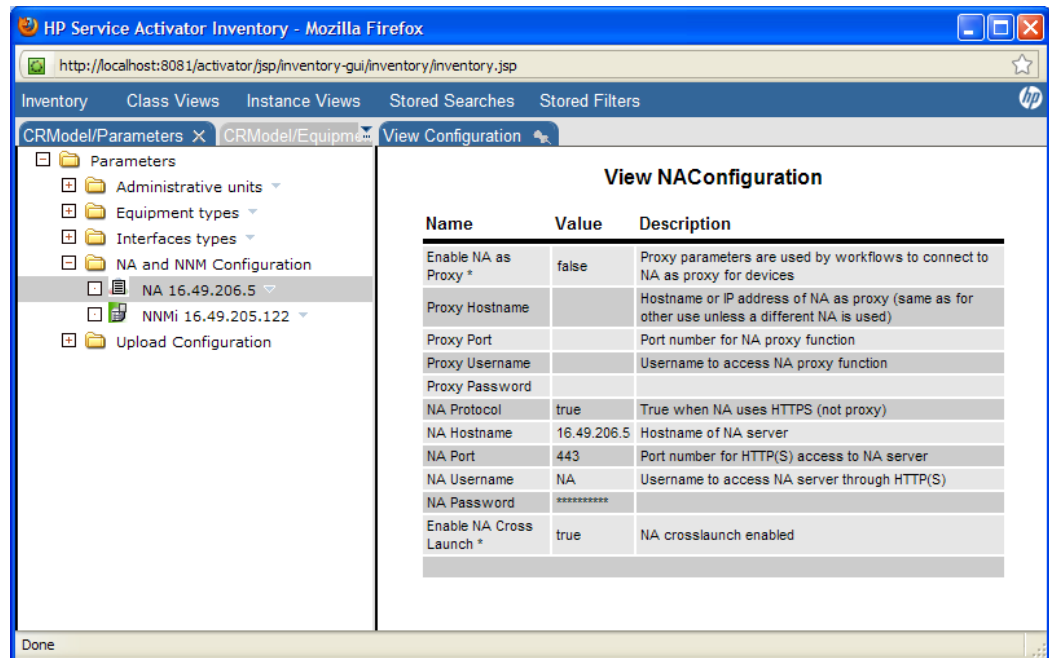
Similarly the NNMi parameters are shown in Figure 4-10 and explained by the comments in the forms.

Figure 4-9   **Parameters Tree NNMi Configuration**
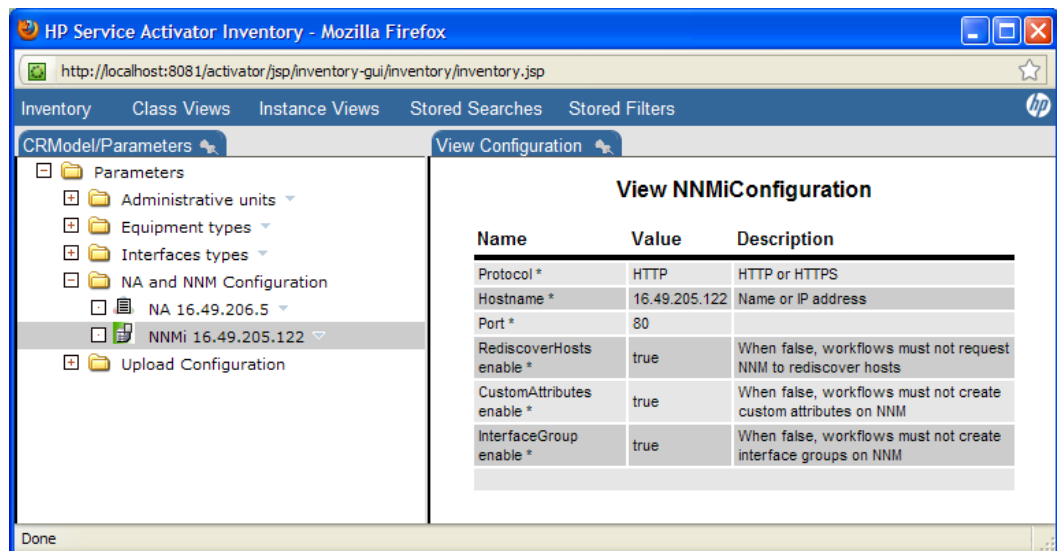
Upload Configuration

This branch has two child branches:

- one for configuring direct interface upload; you can name the queue where informative messages from this operation will be written, and you can set demo mode, implying interface names are read from a file instead of from the device.

- the other one is a subtree for configuring dataload from NNMi; it is described in the following.

The first branch in the subtree defines the scope of the upload. Here you can edit an XML document which lists all the potential devices to be loaded by their hostnames or IP addresses. Devices which are not listed here will be disregarded by the NMMi dataload operation.

The second branch in the NNMi Dataload subtree configures automatic enrichment of the uploaded data to be performed in the first phase of the dataload process. Enrichment functions are also specified in an XML document which can be edited directly in the forms area. It consists primarily of rules. Each rule has a type and a condition which determine when the rule applies, and a Transformation or Assignment which specifies what the rule does. The type identifies the object (resource) class that the rule applies to (Network Element or Interface), the condition can be defined in terms of a value of an attribute of the object loaded from NNMi, see examples in the screenshot in Figure 4-10.

A Transformation is specified as a Java class and method which is applied to the object in question. An Assignment specified a value to assign to an attribute of the object. The example in the screenshot defines (in a comment) some assignments gated by type but no additional conditions, so that in effect these assignments define standard values for fields of dataloaded objects.

Figure 4-10   Parameter Tree Dataload Enrichment
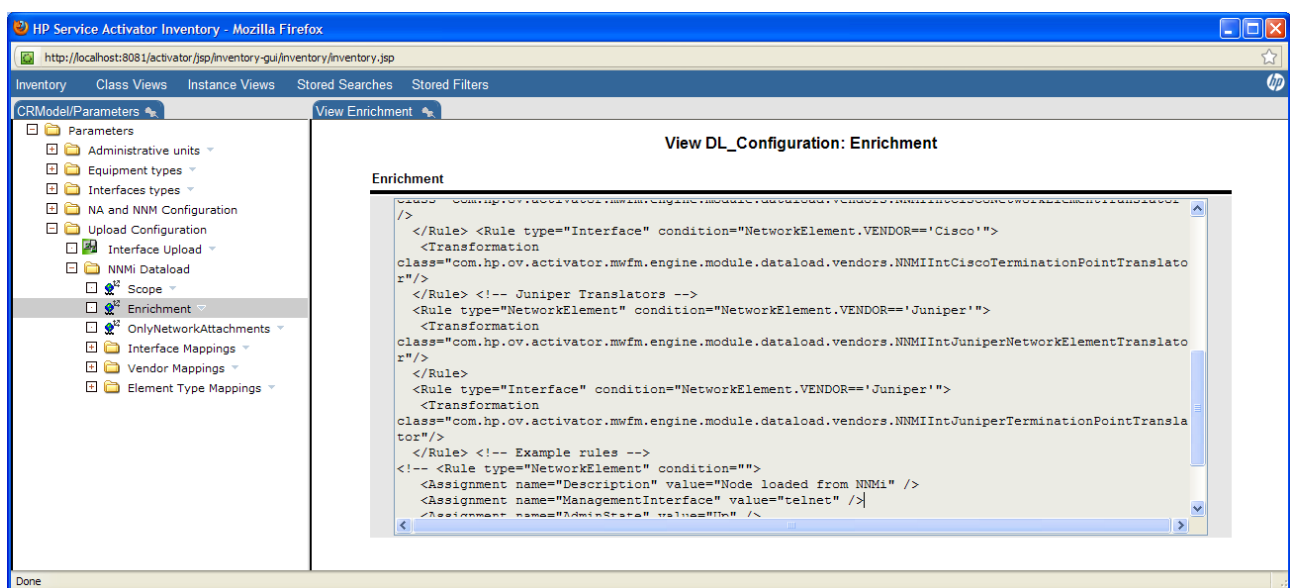
The third branch 'OnlyNetworkAttachment' can be used to disable calculation of the network a network element belongs to and its role, when these items are handled by enrichment rules.

The last three branches of the NNMi Dataload subtree defines how attribute values loaded from NMMi are mapped to interface types, equipment vendors, and element types in the CNRM. The mappings take several attributes of the loaded NMMi objects into consideration and use regular expression patterns to recognize their values. A rich and normally sufficient set of mappings are prepopulated when the CNRM is installed.

5 System Administration User Interface

The Self Management menu of the main UI is shown in Figure 5-1. Administrator privileges are required to access this menu, except for the first item, Change Password. It has a two-level structure with sub-menus. The User Management functions (including Change Password) are described in chapter 6. The functions in the other submenus: Clean Up, Reload, Statistics, Distribution and Export, are described in the following sections within this chapter.

Figure 5-1   Self Management Menu
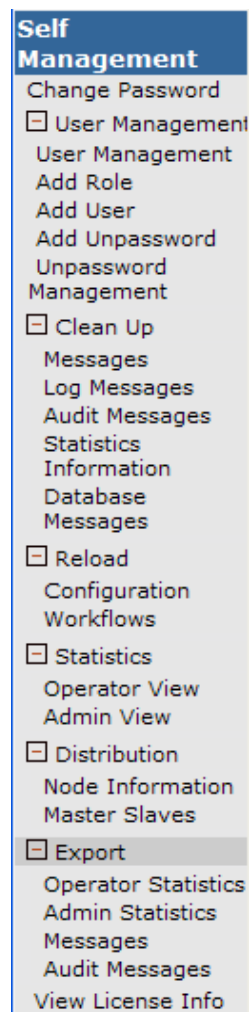
NOTE   If authentication is not enabled for the workflow manager the Change Password item and User Management sub-menu will not appear.

Clean Up

The Clean Up submenu lets you clean up, i.e. delete, information of various kinds that accumulates in the system.

- **Messages** permanently deletes all messages posted on tabs for the Messages view. If you prefer to delete only messages on a single table or even a single message, use right-click actions in the Messages view.

- **Log Messages** lets you delete all inactive logs on a selected host or all hosts.

- **Audit Messages** lets you delete all audit trails records whose creation time is before a time you must specify.

- **Statistics Information** lets you delete all statistics data collected in a specified time interval on a specified host or all hosts.

- **Database Message** lets you delete all temporary database records that have been created by workflows or workflow manager modules in a specified time interval. Normally workflows will do their own cleanup, so this action should not be needed on a regular basis.

NOTE   Be sure not to delete Database Messages which are in use by running workflow jobs.

A system maintenance schedule should include regular cleanup of all these items. Audit trail records are generally of the greatest historical value because they document what the system has done; hence they should only be deleted according to a schedule which may include exporting them for archiving elsewhere.

Reload

The Reload submenu lets you reload information into the running workflow engine. This will update the way the engine works.

- **Configuration** reloads the configuration files for the workflow manager (mwfm.xml) and the resource manager (resmgr.xml) configuration files. Note that some changes cannot be handled by reload, but require a restart of HP Service Activator. Consult the dedicated documentation for the workflow manager and the resource manager for this topic.

- **Workflows** reloads deployed workflow from the workflows repository into the running workflow manager. This will be needed if new workflows are deployed while HP Service Activator is running.

Statistics

Two types of statistical data are collected in Service Activator: server usage statistics and workflow statistics. Collection of statistics is configurable in the workflow manager and must also be enabled in the workflow definition.

NOTE	Workflow statistics is collected only when the auditor module is configured correctly and collection of statistics is enabled at the workflow level.

Service Activator has two types of reports to retrieve and display statistical data: operator and administrator. The Operator Statistical Filter allows you to generate reports on workflows. The Administrator Statistical Filter allows you to generate reports on server usage.

Operator Statistical Filter

The form is accessible through the Operator View in the expandable Statistics item in the Self-Management menu. The form is used to generate statistical reports on workflows according to the parameters you enter. Please see the parameters listed below.

Figure 5-2 **Operator Statistical Filter**

Host: On a single node system, the Host field will be pre-filled with the host name on which ServiceActivator is running. On a multi node system the field will be a drop down list with all the members of the cluster. The Host field will select which cluster node to obtain statistical information for. On a multiple node system it is possible to get accumulated statistics for all systems by selecting All Hosts.

Date and Time. If the "From" field is left empty, Service Activator sets the date to the date of the earliest record found in the list of the selected workflows. If the To field is left empty, the date is set to the date of the latest record found in the list of the selected workflows. You can leave the time fields empty. The default value is 00:00:00 in the From field and 23:59:59 in the To field. It is not possible to complete the time fields without specifying the dates. The date format is available from the tool tips next to the date fields.

In addition to the From and To dates, you can enter the From and To times. However, if you check the Display Chart check-box, remember that the time parameters will be rounded up. Rounding up depends on the Time Unit you select. See the rounding up examples in Table 5-1. The examples in Table 5-1 use the date and time format YYYY.MM.DD hh:mm.

Table 5-1   **Rounding Up Time**

Time Unit	Time Range Entered	Changes of the From Time	Changes of the To Time
Minute	2005.02.11 10:13 — 2005.05.22 12:49	No changes. The date and the time entered will be used, 2005.02.11 10:13:00	Seconds will be increased to 59 for the time entered, 2005.05.22 12:49:59
Hour	2005.02.11 10:13 — 2005.05.22 12:49	The minutes entered will be ignored, 2005.02.11 10:00:00	The time entered will be rounded up to the next hour minus 1second, 2005.05.22 12:59:59
Day	2005.02.11 10:13 — 2007.05.22 12:49	The time entered will be ignored, 2005.02.11 00:00:00	The date and the time entered will be rounded up to the next day minus 1 second, 2007.05.22 23:59:59
Month	2005.02.11 10:13 — 2007.05.22 12:49	The day and the time entered will be ignored, 2005.02.01 00:00:00	The date entered will be rounded up to the next month minus 1 second, 2007.05.31 23:59:59
Year	2005.02.11 10:13 — 2009.05.22 12:49	The date and the time entered will be ignored, 2005.01.01 00:00:00	The date entered will rounded up to the next year minus 1 second, 2009.12.31 23:59:59

If the Display Chart check-box is not checked and only the tables are viewed, the From and To times are used exactly as entered. No rounding up takes place. For instance, if 15:34 is entered as the To time, then it is 15:34:00 (not 15:34:59) that is used.

Workflow Selection. This is the only mandatory parameter in the form. It is possible to select one or more workflows.

Chart Selection. If the Display Chart check-box is checked, then the selected data series is shown. At least one data series must be selected. You can also select the time unit used in the graph when Display Chart is checked. The time unit determines at what intervals data is grouped. You can choose minutes, hours, days, months, and years. The Show Multiple Axes check-box indicates that each data series in the report will have its own axes. Note that workflow activations are always shown along a separate axis. Below you can find the descriptions of the selectable chart graphs.

- **Workflow Duration Time.** The average workflow duration time. It equals the sum of all other times

- **Activation Time.** The average time a workflow spent in the activation phase.

- **Number of Completed Jobs.** The average number of completed workflows.

- **Workflow Processing Time.** The average time a workflow spent on computation.

- **Workflow Persistence Time.** The average time a workflow spent to persist itself.

- **Workflow Wait Time.** This is a calculated value of the average of duration time minus the sum of Activation Time, Processing Time, Persistence Time, and Idle Time.

- **Workflow Idle Time.** The average time a workflow is waiting in a request queue.

Table Columns Selection. Select the columns to include in your report. If no column is selected, tables are not displayed. You can rearrange the order of the columns by selecting a column name clicking [Up] or [Down] buttons.

Administrator Statistical Filter

The form is accessible through the Admin View in the expandable Statistics item in the Utilities menu. It is used to generate statistical reports on server usage. There are no required parameters in the form. Please see the parameters listed below.

Figure 5-3 **Administrator Statistical Filter**

Host. On a single node system, the Host field will be pre-filled with the host name on which ServiceActivator is running. On a multi node system the field will be a drop down list with all the members of the cluster. The Host field will select which cluster node to obtain statistical information for.

Date and Time. If the From field is left empty, Service Activator sets the date to the date of the earliest record found in the list of the selected workflows. If the To field is left empty, the date is set to the date of the latest record found in the list of the selected workflows. You can leave the time fields empty. The default value is 00:00:00 in the From field and 23:59:59 in the To field. It is not possible to complete the time fields without specifying the dates. The date format is available from the tool tips next to the date fields.

The date and the time entered will be rounded up if you check the Display Chart check-box. See Table 5-2 on page 135 for rounding up examples.

Chart Selection. If the Display Chart check-box is checked, then the selected data series is shown. At least one data series must be selected. When Display Chart is checked, you can also select the time unit used in the graph. The time unit determines at what intervals data is grouped. You can choose minutes, hours, days, months, and years. The Show Multiple Axes check-box indicates that each data series in the report will have its own axes. Note that workflow activations are always shown along a separate axis. Below you can find the descriptions of the selectable chart graphs.

- **Worker Threads in Use.** The average number of worker threads running.

- **Activation Threads in Use.** The average number of activations threads running.

- **Running Workflows.** The average number of workflows running.

- **Completed Jobs.** The average number of jobs done.

NOTE   If the Auditor module is not configured in the mwfm.xml and gathering statistics/audit is not enabled for workflows the Workflow Properties window in Workflow Designer, then the number of completed jobs will be shown as 0.

- **Logged in Users.** The average number of logged in users.

Distribution

An HP Service Activator system runs on a platform comprising one or more server hosts, referred to as nodes. For most purposes the distribution of the system across multiple nodes is transparent. The Distribution menu contains functions that allow you to monitor and manage the state of each node. Two views can be launched from this menu: Node Information and Master Slaves. Both of these views will appear in the main working area.

The adminstrator can control the state of host nodes and take them in and out of operation by managing two state attributes of each node: suspended/resumed and locked/unlocked. Locking a node prevents the starting of further jobs, whereas existing jobs are allowed to finish. Suspending a node halts all running jobs until the node is resumed.

Node Information View

This is the view that allows you to monitor and change the state of the nodes of the distributed platform. An example with three nodes, two of them offline, is shown in Figure 5-4.

Figure 5-4   **Node Information View**

Name	Online State	Node Status	Suspend State	Lock State	Takeover Node	Jobs	Activating	Internal Suspend State
elynning	Online	Running...	Resumed	Unlocked				Resumed
jkoenigsfeldt	Offline		Resumed	Unlocked	elynning			
mstenhuus	Offline		Resumed	Locked				
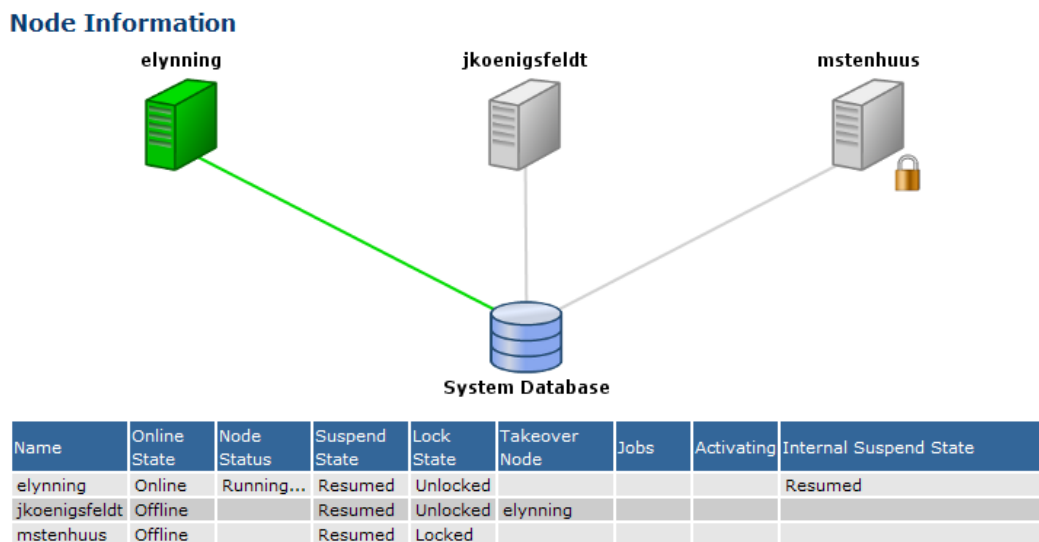
The information that is shown is retrieved from the shared database and by querying the other hosts by the host node that the user is logged in to. The same information - which nodes exist and what are their states - is shown graphically and in list form. The list has one row per host node. The information shown in the columns of the list is:

- Name   Hostname of the node.

- Online State   Online: the node reports itself as running. Offline: the node does not report itself as running, or it is deemed not running by other nodes due to missing heartbeat.

- Node Status   The status of the node as reported by itself: starting up, running or shutting down. When offline, the field will be empty.

- Suspend State   Suspended or Resumed.

- Lock State   Locked or Unlocked

- Takeover Node   For the short period when a node has gone down and its work is being taken over, this field identifies the (other) node that is orchestrating the takeover.

- Jobs   The number of jobs running on the node.

- Activating   The number of ongoing activation tranactions on the node.

- Internal Suspend State   A node can report the internal suspend state when it cannot reach the database (not via the database).

Right-Click Actions for Nodes

Node management actions are selectable per node from a right-click menu. The following actions are available:

In the node Information view, when you right-click on each row, the following State Change Operation (as shown in Figure 5-53) menus items are displayed:

- Suspend Node   Suspend the selected node.

- Resume Node   Resume the selected node.

- Lock Node   Lock the selected node.

- Unlock Node   Unlock the selected node.

- Suspend All Nodes   Suspend all the nodes.

- Resume All Nodes   Resume all the nodes.

- Lock All Nodes   Lock all the nodes.

- Unlock All Nodes   Unlock all the nodes.

NOTE   The actions on all nodes are available only when there is more than one host node in the cluster.

The graph shows all the nodes as icons connected to the shared database. The color code for the icon and connection is as follows:

- green   ok, the node is running and alive

- red   the node is in the internal suspend state (see above)

- blue   the node is suspended

- grey   the node is offline (see above)

In addition to the color codes a padlock is shown close to a node that is locked.

Virtual IP Address Failover

At system installation time it is possible to associate a virtual IP address with each cluster node in addition to its hard IP address. If the cluster node fails, its virtual IP address will migrate ("fail over") to (one of) the remaining operational node(s) in the cluster. The advantage of the virtual IP address is that if a client system uses the virtual IP address of a cluster node to send request messages to HP Service Activator, it may continue to send requests to this address even after the node fails. In the Node Information view the virtual IP addresses are shown against each node, and after a failover the address of the failing node (which is greyed) is shown in red against the node that it has failed over to. This is illustrated in Figure 5-5.

Figure 5-5   **Node Information View, Virtual IP Address Failover**

Node Information

Primary site: PRIMUS										
Name	Online State	Node Status	Suspend State	Lock State	Takeover Node	Jobs	Activating	Internal Suspend State	Virtual IP	Taken Over Virtual IP
msh1.hpsa	Offline		Resumed	Unlocked					16.49.200.203	
msh2.hpsa	Online	Running...	Resumed	Unlocked					10.10.10.111	eth0:2#16.49.200.203
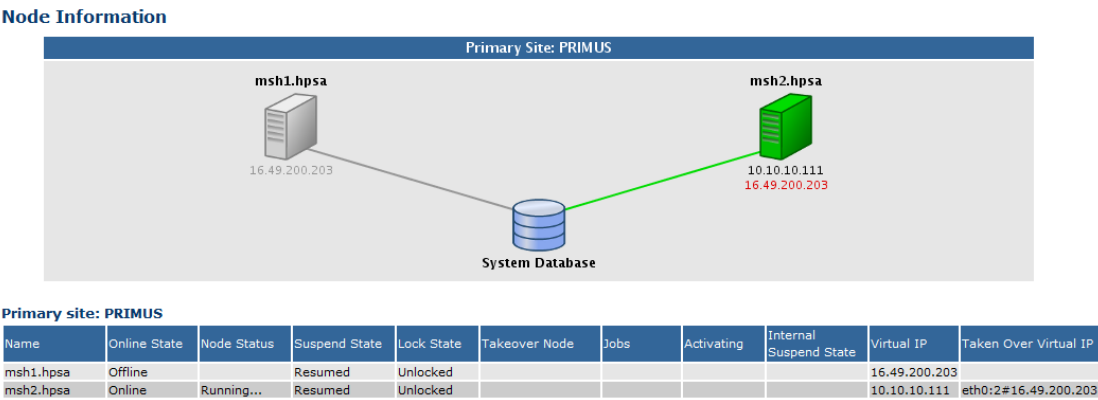
Disaster Site

It is also possible at HP Service Activator installation time to configure that nodes belong to sites, and that one or more sites are standby sites. Workflow jobs will not be assigned to a node at a standby site.

If a disaster strikes, so the nodes and database servers at the primary site become unusable, a standby site can be reconfigured to become primary. This can be done - typically from a browser at the standby site - by right-clicking in the Standby Site part of the graphic in the Node Information view; see Figure 5-6. A popup window will appear to allow you to change the state of the site. The graphic shows the normal situation where the Primary site is operational.

Figure 5-6   **Node Information View, Primary and Standby Site**

Node Information

Primary site: PRIMARY								
Name	Online State	Node Status	Suspend State	Lock State	Takeover Node	Jobs	Activating	Internal Suspend State
JOHNDOE	Offline		Suspended	Unlocked				
MSTENHUUS	Online	Running...	Resumed	Unlocked				

Standby site: DISASTER								
Name	Online State	Node Status	Suspend State	Lock State	Takeover Node	Jobs	Activating	Internal Suspend State
BACKUP	Offline		Suspended	Unlocked				

Master Slaves View

The scheduler module of the workflow manager must not be active on more than one node, which will known as the master. The passive nodes are called slaves. In this view you can configure which node is the master. The view has a general form, prepared for the addition of further modules that need to operate in this mode.

Export

Information of several kinds can be exported from HP Service Activator's repositories to csv format which you can open in a spreadsheet or save in a file. In each case you must specify in a popup window which records to export in terms of the period the data originates from from start to end time, the source host (a specific one or all hosts), and possibly additional criteria depending on the kind of data. When you then press the [Submit] button, you will proceed to standard popup windows as used for downloading to specify whether to open or save the data, and in the latter case the name and directory path of the file.

NOTE	When you choose to open exported data the spreadsheet will occupy the browser window that it was launched from.

All exporting of data is done from the Export submenu. The menu has four items:

- **Operator Statistics** exports statistics collected from workflows

- **Admin Statistics** exports statistics about server usage

- **Messages** exports messages (as shown in Messages view). You can select to export from a single tab (queue) or all tabs.

- **Audit Messages** exports audit records. You can select to specify a single event type or leave it blank to export all event types,

6   User Management

In the Self Management menu in the main HP Service Activator user interface window there is an item, `Change Password`, which any operator can use to change his/her own password for use with the native authentication module, and the User Management sub-menu which is only available to the system administrator.

With the functions that can be selected in the User Management sub-menu it is possible to assign inventory UI privileges to roles regardless of which authentication module is used and to manage teams and users, their roles and authentication details for HP Service Activator's native authentication module (`DatabaseAdvancedAuthModule`). The User Manangement sub-menu has five selections as shown in Figure 6-1 and described below. The first one, User Management, is the one with the most functions and is described last.

There is a predefined system user whose username and password is defined at system installation time. This user never expires and has all roles and privileges, so it is handy to use it for the system administrator.

It is optional to organize users in teams. This feature, which must be enabled in the configuration of the native authentication module, allows the system administrator to delegate user management to team administrators. The roles that can be used by members of the team must be assigned by the system administrator to the team as a whole. The team administrator can then add users as team members and assign roles which the team has to the individual members. If you don't define teams, all users will belong to a predefined default team.

Passwords are entered when a user is created or the user is updated, or when a user changes his/her own password. The native authentication module can be configured to validate that passwords are not too simple: they must then be at least 8 characters long, and must include at least one of each of these: upper and lower case letters, digits and special characters. You can define strings that will not be accepted as passwords ("unpasswords").

Figure 6-1   **User Management Sub-menu**
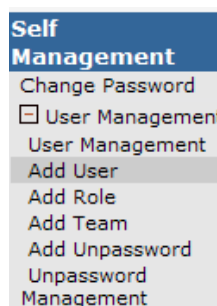
Add User

Here you can create a new user (remember: only for the native authentication module). The details are entered in a simple form in a pop-up window as shown in Figure 6-2. If password validation is

enabled, passwords must satisfy the rules described above. The description, real name and company name are purely informative for display within User Management.

If teams are used, a team administrator can create new users, but only as members of the same team. A system administrator can create new users in all teams and make them team administrators. If you do not use teams, just leave the default team.

Note that you have options to force the user to change the password on first login and prevent password expiration.

Figure 6-2   **Add User Form**

Add Role

Here you can create a new role. You cannot assign a role to a user without first creating the role. If you want to assign privileges for operations on the inventory UI to the role, you must create it, even if you use a different module for authentication. The details are entered in a simple form in a pop-up window.

NOTE

If you use an authentication module different from the native one, authorization of user roles is not based on the database managed through the User Management interface described in this chapter. In these cases roles must still be created here, but only to maintain the relationship between roles and inventory UI privileges (wou will not need to create roles that are only used for workflows).

The other available authentication modules authorize roles based on the operating system user groups that a user belongs to; see "Authentication and Assigning Roles to Users" in chapter 7 of "HP Service Activator; System Integrator's Overview".

Add Team

Here you can create a new team. When, as system administrator, you have created a new team, you will normally also create a user as team administrator. Until you do that, only the system

administrator can manage the users in the team. You will also need to assign the proper roles to the team before they can be assigned to the team members, typically by the team administrator.

NOTE	The teams feature is only available when configured (in file `mwfm.xml`)

Add Unpassword

Here you can define a string that will not be accepted as a password in the definition of a user, or when a user changes his own password.

Unpassword Management

Here you can search the dictionary of unacceptable passwords. Enter a substring, and all matching entries will be returned. If you leave the field empty, all entries will be retrieved. You can right-click on a row in the list of search result and execute an action to delete it.

User Management

This item selects the main view for user and role management. The view has eight lists selected by tabs: User Info, Role Info, Team Info, Operation Type, Branch Type, Tree Definition, Filters and Searches. They are described here in reverse order.

Searches

Advanced searches are described in chapter 3. The definition of an advanced search in terms of object attribute values and comparison operators can be saved with a name.

This tab shows a list with names and descriptions of stored inventory searches. Initially when a search has been stored it can be used by the user who stored it, and it will then be shown in the list. As administrator you can make searches which have been stored by one user available to other users as well. To do this, right-click on a search and select the function to update its associations to users.

Filters

Inventory filters are described in chapter 3. The definition of a filter in terms of values for filter variables can be saved with a name.

This tab shows a list with names and descriptions of inventory filters. Initially when a filter has been stored it can be used by the user who stored it, and it will then be shown in the list. As administrator you can make filters which have been stored by one user available or mandatory to other users as well. Mandatory means the filter is applied automatically. To do this, right-click on a filter and select the function to update its associations to users.

Operation Type, Branch Type, Tree Definition

The items on these tabs are the operation privileges defined for the inventory UI. All three are read-only lists showing you the privileges that have been defined for inventory UI trees. Operation Types are privileges to perform operations on branches in trees. Branch Types are privileges to view and access branches. Tree Definitions are privileges to view trees; there is one for each tree defined. You can browse them to understand which inventory UI privileges exist.

Team Info

This tab is where you can assign roles to teams. The displayed lists shows existing teams; see Figure 6-3 for an example.

Figure 6-3   Team Info List
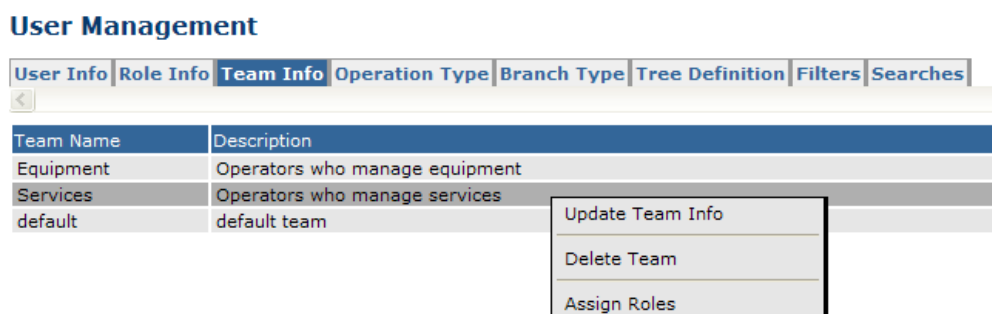
The menu appears when you right-click on a team in the list. Select an item from the menu, and a window will pop where the operation can be performed:

- **Update Team Info**   the name and description of an existing team can be modified.

- **Delete Team**   a team can be deleted.

- **Assign Roles**   roles can be assigned to or removed from a team; the window is similar to the one shown below for assigning roles to a user.

Role Info

This tab is where you can assign privileges for inventory operations to roles. The displayed list shows existing roles; see Figure 6-4 for an example.

Figure 6-4   Role Info List
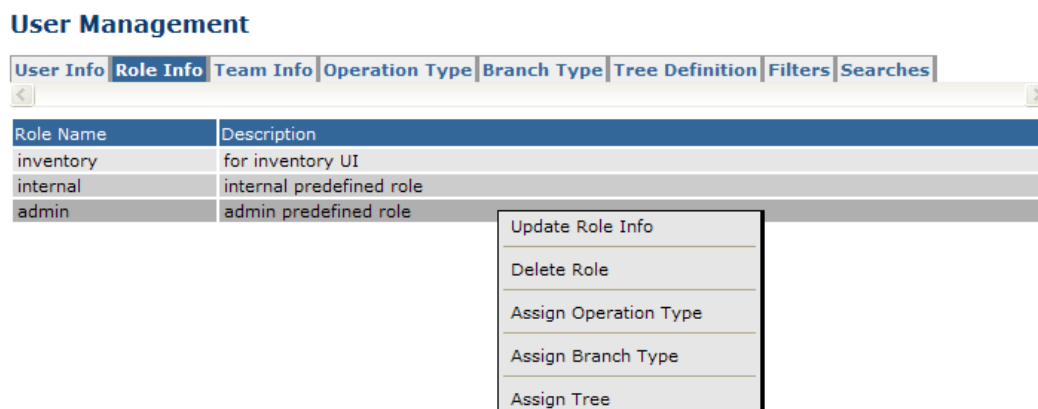
The menu appears when you right-click on a role in the list. Selecting an item from the menu pops up a window to perform the indicated operation:

- **Update Role Info**   the name and description of an existing role can be modified.

- **Delete Role**   a role can be deleted.

- **Assign ...**   individual privileges of the indicated types can be assigned to or removed from a role; the window is similar to the one shown below for assigning roles to a user.

User Info

This list is for management of existing users. See Figure 6-5 for an example.

NOTE   You can only manage users with the User Info list when the native authentication module (`DatabaseAdvancedAuthModule`) is used. When another authentication module is used (as configured for the workflow manager) the User Info tab is not available. How users are then managed depends on the selected authentication module; see *HP Service Activator, System Integrator's Overview* for more information.

Figure 6-5   **User Info List**

NOTE   The system user is predefined with name and password established during installation of HP Service Activator. It cannot be disabled. It exists for technical implementation reasons and has all roles including the admin role. This allows you to log in and create a proper administrator user.

Right-click on a user row to get the menu of operations you can perform to manage a user. When you select an item from the menu a window pops up where you can perform the operation as follows:

- **Update User Info**   the data about the user can be modifed. The window is similar to the one for Add User.

- **Copy User**   similar to the Add User function, but with non-personal fields prefilled with value from the right-clicked user.

- **Assign Roles**   roles can be assigned to the user, see Figure 6-6.

- **Delete User**   a user can be deleted.

- **Enable/Disable User**   the user can be enabled or deleted, the item which appears depends on the current state. A disabled user will be refused authentication.

Figure 6-6   **Role Assignment Window**
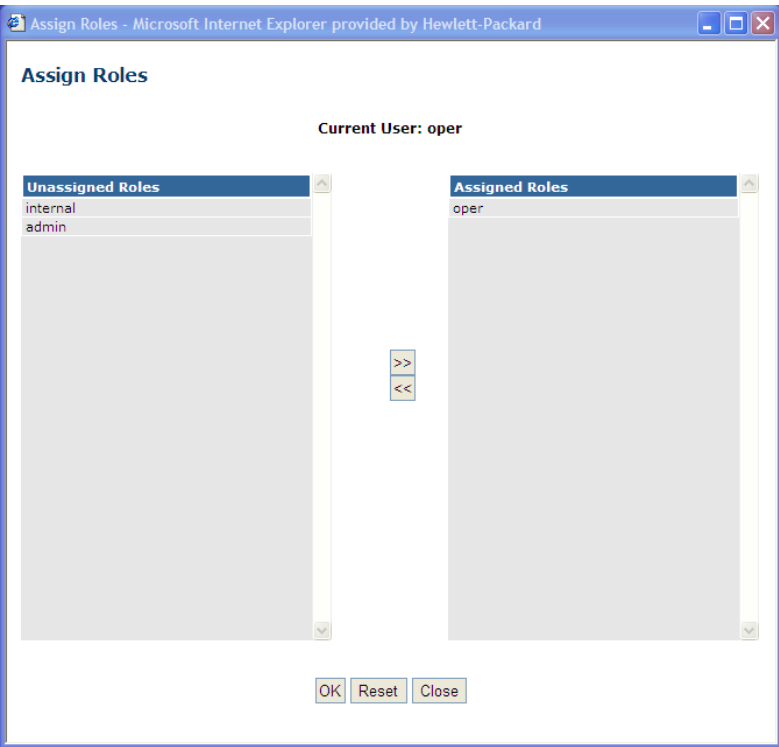
NOTE   The roles admin and internal are predefined. The admin role carries system administrator privileges. The internal role exists for technical implementation reasons and should normally not be assigned to human users.

7 System Configuration

Normally the configurations described here will be performed by the solution delivery team. This documentation is provided here to enable a system administrator to understand how properties of the user interface are configured and, if necessary, modify the configuration.

As the last step of HP Service Activator software installation the script ActivatorConfig is run. This script creates the basic database tables used by HP Service Activator and a number of system configuration files. Subsequently the solution can be further configured by editing the configuration files. ActivatorConfig also configures a number of special features: single sign on, site setup as primary or backup for disaster recovery, use of secure sessions (ssh) by plug-ins. All configuration done with ActivatorConfig is described in the Installation Guide (different versions apply depending on operating system). You can rerun ActivatorConfig also after installation has been completed, but you must then be aware that the configuration files for the workflow manager and HP Service Activator's user interface are reset in the process, so you must save them before running ActivatorConfig and restore/reconstruct them after.

All configuration files for the HP Service Activator system are listed and briefly described in Appendix B. These files are generally richly commented so that one can work with them without making reference to documentation. This chapter is intended primarily to help you understand what can be configured.

The main components to configure are the workflow manager and the user interface. This chapter is primarily focused on the user interface, as the workflow manager is assumed to be configured as part of solution installation.

Note that some functions associated with the Common Network Resource Model are configured in database tables. This includes parameters for integration of HP Service Activator with NNMi and NA and is described in 4

Work Area and Tools Menus

The contents of the WorkArea and Tools menus in the navigation pane of the operator UI are configurable. The default contents of these menus, as described in chapter 2, under "HP Service Activator Main UI Window", is defined in the file $ACTIVATOR_ETC/config/menu.xml (there is nothing for the Tools menu, the Refresh item is special and not configurable). It is recommended not to change this file, but to define modifications in additional files that must be placed in the $ACTIVATOR_ETC/config/menus directory. A separate file should be used for each solution, and will typically be installed as part of the solution. It is recommended to name the file after the solution: <solution name>_menu.xml.

By configuration it is possible to remove items or modify items from the default menu and to add new items. A role can be associated with a new or modified menu item, so that only users who possess that role will see the item in their menu. The default menu configuration does not mention any roles.

Each menu item is defined primarily by the label which is displayed in the menu and the the Java Server Page which is launched when the item is selected.

It is possible to define one level of submenu, i.e. a labelled item that can be opened to show a list of child menu items but is not itself selectable for launch.

Solution menu files must be XML formatted according to the definition file
$ACTIVATOR_ETC/config/menus/solutionmenu.dtd.

Each item to be selectable in the menu is defined by an <Item> element. All <Item> elements are enclosed by the outermost <Menu> element and its child <WorkArea> or <Tools> element, like this:

```
<Menu>
  <WorkArea>
        <Item> ... </Item>
                ...
        <Item> ... </Item>
  </WorkArea>
  <Tools>
        <Item> ... </Item>
                ...
        <Item> ... </Item>
  </Tools>
</Menu>
```

The <Item> element has an Operation attribute which takes one of the values ADD (default), DELETE or MODIFY, and some of the following child elements (at most one of each):

<Id>   Mandatory, has a numeric value which identifies the item for cross-reference and determines the order in which menu items are shown. With DELETE or MODIFY it identifies the default menu item to delete or modify.

<Label>   Mandatory for ADD, optional on MODIFY. The string to be displayed in the menu.

<Role>   Optional. Name of a role the user must have for the item to be available in the menu.

<Parent>   Only used for an item belonging to a submenu, equals value of <Id> for the item representing the submenu.

<Url>   Mandatory for ADD, optional for MODIFY, not used on a submenu item. The value identifies the page to be launched for the item, typically in the form of a JSP file. Page files are identified within and relative to the root of the *JBOSS_DEPLOY*/hpovact.sar service archive. Typically they are stored in the activator.war/jsp directory. If file sample.jsp is stored there, it will be identified by the value /activator/jsp/sample.jsp.

           The <Url> element has a number of attributes to control the way the JSP is launched and presented in a frame; they are explained in Table 7-1.

Table 7-1   UI Menu Item Attributes

Attribute name	Value	Default	Description
popup	true | false	false	If true, the page is shown in a pop-up window; if false, it is shown in the work area of the main UI window regardless of any other attributes.
multiple	true | false	false	If true, a new pop-up window is always launched; otherwise if there is already an open pop-up window for this item it is focused and reused.
location	true | false	false	If yes, the pop-up window will have a location bar.
menubar	true | false	false	If yes, the pop-up window will have a menu bar.
resizable	true | false	true	If yes, the pop-up window will be resizable.

Attribute name	Value	Default	Description
scrollbars	true | false	true	If yes, the pop-up window will have vertical and horizontal scrollbars.
status	true | false	false	If yes, the pop-up window will have a status bar.
width	number	700	The pop-up window is opened with the specified width (pixels).
height	number	600	The pop-up window is opened with the specified height (pixels).
left	number	150	Distance (pixels) from left side of frame to the page.
top	number	150	Distance (pixels) from top of frame to the page.

Other User Interface Configuration

Apart from the work area menu as described above, all configurations for the user interface are in file *$JBOSS_DEPLOY*/hpovact.sar/activator.war/WEB-INF/web.xml. The location of this file is due to the fact that the user interface is deployed as a service under JBoss using general web service features of JBoss.

The file is an XML document. All the parameters are defined as (name, value) pairs. We refer to the parameters here by their names, which appear in <param-name> elements. The corresponding value is stated in the immediately following <param-value> element.

The web.xml file is read only at system startup time, hence you will need to restart HP Service Activator when you have changed it.

Parameters which a system administratot may wish to modify are listed and described in Table 7-2.

Table 7-2   **User Interface Parameters**

Parameter	Description
job_refresh_rate	Refresh rate in seconds for most lists shown in the main working area. Default is 45.
track_refresh_rate	Refresh rate in seconds for the Track Activations view. This is not covered by job_refresh_rate, because you are likely to want it to be faster, so you can watch the view change. Default is 10.
counter_refresh_rate	Refresh rate in seconds for job counters in navigation pane. Default 4. Refreshing the counters may require a lot of processing. Increase the rate if there are more than 10 users.
job_records_max_records	Standard page size ("Retrieve limited jobs") for Active Jobs view. Default is 200.
audit_messages_max_records	Page size for Audit Records view. Default is 60.

Parameter	Description
messages_max_records	Page size for Messages view. Default is 1000.
services_max_records	Page size for Services view. Default value is 200.
queue_hidden_<Queue>	This parameter only exists if you create an <init-param> element to define it. If the value is true, the queue with name <Queue> (to appear in the queue tab) will be hidden.
queue_interactible_<Queue>	Generally, this parameter only exists if you create an <init-param> element to define it. It affects jobs listed under the queue tab named <Queue>. If the value is false, the right-click menu action Interact with Job will not appear for such jobs. When a queue is intended for synchronization between workflows, it makes sense to disable interaction by users.
job_field_order	Defines columns appearing in lists of running and queued jobs in the Active Jobs view. Default is: Job Id, Service Id, HostName, Workflow, Status, Start Time, Post Time, Step, Description. Also available are Order Id, (workflow) Type, (workflow) State, Trace Role.
job_field_order_<Queue>	Defines columns appearing in job list under the queue tab named <Queue>. Overrides the general definition by job_field_order.
casepacket_name<n> casepacket_label<n>	These parameters must appear in pairs with matching values for <n>, which must be 1, 2, etc., for different pairs. The name is used to match a workflow job (case-packet) variable. The label can be used to define a column in a job list (job_field_order and job_field_order_<Queue>). In this way you can pick any variable from workflow jobs and display it in job lists.
audit_field_order	Defines columns appearing in Audit Records view. Default is: Job Id, Service Id, Order Id, Workflow Type, Workflow State, HostName, Workflow, Date, Event Type, Step. Additional options are: Message, Class, Username.
sov_query_fields	Defines which query fields will be present in the Service Order View. Must be coordinated with Query-Filter-Order parameter in mwfm.xml which defines the order in which the queries are applied: do not mention in Query-Filter-Order a parameter which is not present.

Parameter	Description
job_field_order_sov	Defines columns appearing in lists of running and queued jobs in the Service Order view. The following columns can be used: sov_job_id, sov_service_id, sov_order_id, sov_hostname, sov_workflow, sov_start_time, sov_type, sov_state, sov_step. You can reorder the columns and drop some of them, but you cannot add more columns.
job_field_order_sov_scheduled	Defines columns appearing in lists of running and queued jobs in the Service Order view. The following columns can be used: sov_job_id, sov_service_id, sov_order_id, sov_workflow, sov_type, sov_state, sov_start_time, sov_repetition. You can reorder the columns and drop some of them, but you cannot add more columns.
sov_message_fields	Defines columns appearing in message lists in the Service Order view. The following columns can be used: sov_job_id, sov_queue, sov_service_id, sov_order_id, sov_hostname, sov_workflow, sov_post_time, sov_type,sov_state, sov_step, sov_message. You can reorder the columns and drop some of them, but you cannot add more columns.
sov_audit_fields	Defines columns appearing in the audit record lists in the Service Order view. The following columns can be used: sov_job_id, sov_identifier, sov_hostname, sov_workflow, sov_order_id, sov_date, sov_type, sov_state, sov_event_type, sov_step, sov_message. You can reorder the columns and drop some of them, but you cannot add more columns.
sov_active_tx_fields	Defines columns appearing in the active transactions lists in the Service Order view. The following columns can be used: sov_job_id, sov_service_id, sov_order_id, sov_workflow, sov_act_start_time, sov_priority, sov_atomic_task, sov_lock_status, sov_lock_arguments, sov_hostname, sov_tx_state. You can reorder the columns and drop some of them, but you cannot add more columns.
sov_job_max_records	Defines maximum number of jobs to be shown in a list (running, scheduled or queued) in the Service Order View.
sov_message_max_records	Defines maximum number of messages to be shown in the Service Order View.
sov_audit_max_records	Defines maximum number of audit records to be shown in the Service Order View.

Parameter	Description
sov_tx_max_records	Defines maximum number of active transactions to be shown in the Service Order View.
messages_read_count_ frequency	Defines how often to read size of tables of messages, audit records and db messages to decide if info or warning icon shall be shown as part of System Status. The read occurs along with counter refresh; when the value here is n, it happens every n^{th} time the counters are refreshed.
messages_warning_level	Thresholds triggering info or warning in System Status for messages (Messages view).
audit_warning_level	Thresholds triggering info or warning in System Status for audit records (Audit records view).
db_messages_warning_level	Thresholds triggering info or warning in System Status for db messages (temporary working memory for workflows).
business_calendar_time_ format	Defines format for hours of working days: 12hr (AM/PM) or 24hr.
priority_value<i> priority_name<i>	These pairs define names for priority values for use in the Change Priority (of workflow job) popup window. For <i> use a value in the range 1..10.
auto_logout_time	Timeout (seconds) for inactivity within a user session before the session is terminated.
auto_logout_warning_time	This long before the session inactivity timeout expires a warning is shown.

Workflow Manager

Many functions of the workflow manager must be enabled by configuring a so-called workflow manager module to perform them. These functions include:

- authentication of users at login time including validation of password composition, password expiration and reuse policy.

- writing of audit trail records

- listening for messages from northbound systems or other external systems on sockers or otherwise

- sending messages to external systems

- collecting workflow and server statistics

More information about configuration of the workflow manager is given in *HP Service Activator, Workflows and the Workflow Manager.*

Appendix A   Scripts

This table contains locations and descriptions of the scripts that are available in Service Activator. Unless otherwise indicated, these files are located in the $ACTIVATOR_BIN directory.

Table A-1 **Service Activator Scripts**

Script	Description
/etc/init.d/activator	Starts/stops the Service Activator processes (Solaris).
/sbin/init.d/activator	Starts/stops the Service Activator processes (HP-UX).
ActivatorConfig[.bat]	Configures Service Activator for a specific environment.
AssignNonRoot	Configures Service Activator to run as non-root, only on UNIX
CatchSocketSenderMessages[.bat]	Listens for messages on a given port and prints those messages to stdout. This script is typically used for testing and demonstration of the SocketSenderModule of the Workflow Manager. By default, it listens on port 4099, but takes a single parameter to specify the port.
checkLicence.[bat]	Checks the status of the HP OpenView AutoPass license and prints out debug information.
CleanLogs[.bat]	Deletes all but the active logs
crypt[.bat]	Encrypts or decrypts a password for local use, to avoid storing unencrypted passwords in the workflow manager configuration file.
DataSourceConfiguration[.bat]	Assists in management of data source configurations.
dc[.bat]	Starts Data Collector, command line tool for gathering information about the Service Activator components (Workflow Manager, Resource Manager, JBoss).
DeleteCompleteTransactions[.bat]	Cleans up saved completed activation transactions.

Script	Description
deploymentmanager[.bat]	Invokes the Deployment Manager executable.
designer[.bat]	Runs the Workflow Designer tool
generateEncryptedPassword[.bat]	Utility to generate an encrypted password. This can be used when an additional data source file has to be created.
generateMD5[.bat]	Calculates MD5 checksum for a file.
InventoryBuilder[.bat]	Runs the InventoryBuilder tool.
InventoryTreeDeployer[.bat]	Runs the InventoryTreeDeployer tool.
InventoryTreeDesigner[.bat]	Runs the InventoryTreeDesigner tool.
modifySystemPassword[.bat]	Utility to update the system user password.
mwfmtool[.bat]	A command line tool for performing workflow engine tasks such as starting workflows and viewing posted messages. If this script is executed without any parameters, it will display a list of all the tasks that can be performed.
remove.serviceactivator	Uninstalls Service Activator on UNIX.
servicebuilder[.bat]	Invokes the Service Builder executable, either the command line (if arguments are passed) or the GUI (if no arguments are passed).
TestAtomicTask[.bat]	Starts an atomic task for testing purposes.
UMMData[.bat]	Imports/export roles with inventory UI privileges from/to file.
updateLicence.[bat]	The script lets you update your trial or existing licence for HP Service Activator.
ViewTransactionState[.bat]	Displays the different states of a completed transaction.
WebServiceDesigner[.bat]	Runs the WebServiceDesigner tool.

Some files with examples of scripts for use with Oracle are provided in the directory
$ACTIVATOR_OPT/examples/database/oracle10 as listed in the following table.

Table A-2   **Example Files for Oracle**

File	Description
`create_ovdb_ora.sh`	Example of files used to create an Oracle 10g database instance.
`listener.ora`	Example of an Oracle listener configuration file used to enable remote database access.
`tnsnames.ora`	Example of an Oracle local naming parameters file used to define aliases for referencing Oracle databases.

Appendix B   Configuration Files

The following table identifies and describes the configuration files that are provided with Service Activator. Unless otherwise indicated, these files are located in *$ACTIVATOR_ETC*/config.

Table B-1   **Service Activator Configuration Files**

File	Description
`CompoundTask.dtd`	Document type definition (DTD) for compound task files created by Service Builder.
`designer.xml`	Configuration file for the Workflow Designer.
`inventoryTree.dtd`	DTD for inventory tree definition file.
`mwfm.xml` `(mwfm.dtd)`	Configuration file for the Workflow Manager and associated DTD.
`par.dtd`	DTD for the MANIFEST/par.xml file found in the Plug-in Archive (PAR).
`resmgr.xml` `(resmgr.dtd)`	Configuration file for the Resource Manager and associated DTD.
`role_mappings.xml` `(role_mappings.dtd)`	Definition file and associated DTD for configuring role mappings. The role mappings definition file is optional.
`service_builder.xml`	Configuration file for Service Builder.
$ACTIVATOR_ETC/ `workflows/workflow.dtd`	DTD for workflow definition.
$JBOSS_DEPLOY/ `hpovact.sar/META-INF/` `jboss-service.xml`	Contains the classpath for the Service Activator J2EE components deployed in JBoss.
$JBOSS_DEPLOY/ `hpovact.sar/activator.war/` `WEB-INF/web.xml`	Configuration file for the UI and servlets. For additional information, see the following references: Chapter 7, the section "Other User Interface Configuration" in this manual; Chapter 4, the section "Adding a Data Source for Inventory UI" and Chapter 5, the beginning, in *HP Service Activator, Inventory Subsystem*; Chapter 4, the section "AskFor" in *HP Service Activator, Workflows and the Workflow Manager*.

File	Description
$JBOSS_DEPLOY/ hpovact.sar/deployer.war/ WEB-INF/web.xml	Contains the configuration for the deployer servlet used by Service Builder. See "Configuring Authentication or Authorization" in *HP Service Activator, Developing Plug-Ins and Compound Tasks* for a description of the configurable parameters in this file.
$JBOSS_DEPLOY/ hpovact-inventory-ds.xml	JBoss data source file containing the database configuration used by the UI.
$JBOSS_DEPLOY/ hpovact-ui-ds.xml	JBoss data source file containing the datasource configuration used by the UI.
dm.dtd (dm.xml)	Configuration file and associated DTD for the Deployment Manager.
deploy.dtd	DTD for deployment descriptor.
$JBOSS_DEPLOY/ mwfm-default-ds.xml	JBoss data source file configuring connections used by the default ("db") database module in the Workflow Manager.
$JBOSS_DEPLOY/ resmgr-default-ds.xml	JBoss data source file configuring connections used by the Resource Manager.
$JBOSS_HOME/ server/default/conf/ login-config.xml	The JAAS login configuration file. It is here that the information about user name and encrypted password for datasources are configured.
$JBOSS_DEPLOY/ hpovact.sar/ hpovact-EJBs.jar/ META-INF/jboss.xml	The MaximumSize setting configures the maximum number of concurrent invocations of the EJB that performs task activations. For additional information, see Chapter 4, "Configuring Activation Parameters," in *HP Service Activator, Developing Plug-Ins and Compound Tasks*.
$JBOSS_DEPLOY/hpovact.sar/ hpovact-EJBs.jar/ META-INF/ejb-jar.xml	Contains the activation time-out configuration parameter. See Chapter 4, "Configuring Activation Parameters," in *HP Service Activator, Developing Plug-Ins and Compound Tasks* for additional information.
$JBOSS_DEPLOY/ hpovact-ra-ds.xml	The max-pool-size setting configures the maximum number of instances of the resource adapter that can be used during task activations. For additional information, see Chapter 4, "Configuring Activation Parameters," in *HP Service Activator, Developing Plug-Ins and Compound Tasks*.
$JBOSS_DEPLOY/ hpovact-oracle-ds.xml	JBoss data source file configuring connections used to access the task repository.
menu.xml (menu.dtd)	Default menu definition file for Work Area menu and its DTD.
solutionmenu.dtd	DTD for solution specific modifications to menu definition.

Appendix C   Log Files

This appendix contains the names and descriptions of the log files used by Service Activator.

Service Activator uses several log files to track events that occur within the product. Many of the logs use an XML-based grammar (those with an `.xml` suffix, listed in the following table). These log files roll over to a new log file once they reach a specific size. You can configure the log size by setting the `log_max_entries` parameter in `mwfm.xml`, `resmgr.xml`, and `hpovact-ra-ds.xml`.

The log files are not localized.

All log files are visible in the Operator UI, where you can view the contents of each log file and manage the log files by removing them periodically. When a log reaches a specific size, Service Activator renames the log by appending `_<file#>.xml` to the log name. Service Activator increments the file number each time it rolls over a log for that component. For example, when the `mwfm_active.xml log` file reaches a specific size, it rolls over to `mwfm_0.xml`.

The actual log files are located in the `$ACTIVATOR_VAR/log/<HOSTNAME>` directory, where HOSTNAME is the computer name on which the Logs are running. Log files ending in `_active.log.xml` are the active logs for the different Service Activator components. These are the only log files that are refreshed when the Auto-refresh feature is turned on and the user is currently viewing this file. Removing these logs when the component is active can cause Service Activator to fail.

Table C-1 lists the log files used by HP Service Activator.

Example C-1 shows the XML DTD for HP Service Activator log files, and Example C-2 shows a typical .xml log entry.

Table C-2 lists a couple of JBoss log files that may be useful when working with Service Activator. These log files can be found in `$JBOSS_HOME/server/default/log`.

Table C-1   **Service Activator Log Files**

File	Description
`connector_active.log.xml`	Provides log entries for transactional coordination of an activation. Also see the `resmgr_active.log.xml`.
`depeng_active.log.xml`	Provides log entries for all deployments of PARs (Plug-in Archives) and compound tasks.
`mwfm_active.log.xml`	Contains the log entries posted by the Workflow Manager during normal operation.
`resmgr_active.log.xml`	Contains the log entries posted by the Resource Manager during normal operation.
`designer.stdout`	Contains stdout diagnostics from running the Workflow Designer. Useful when trying to identify problems running the Workflow Designer.
`designer.stderr`	Contains stderr diagnostics from running the Workflow Designer. Useful when trying to identify problems running the Workflow Designer.

Example C-1 DTD of XML-based Log Files for Service Activator

Example C-1   **DTD of XML-formatted Log Files for HP Service Activator**

```
<!-- The log entry -->
<!ELEMENT LogEntry (Time, Module, Part, Component, Topic?, Thread, ID?,
Message)>
<!-- Level of the entry. It can be one of INFORMATIVE, WARNING, ERROR, DEBUG,
DEBUG2 -->
<!ATTLIST LogEntry level CDATA #REQUIRED>
<!-- The host where the component was running -->
<!ATTLIST LogEntry machine CDATA #IMPLIED>
<!-- Date and time this entry was written -->
<!ELEMENT Time (#PCDATA)>
<!-- Specifies which major module of Service Activator generate the log entry.
The possible values for this field are: MWFM, Connector, RM, Inventory, GUI,
DeploymentEngine, LOGGER_THREAD. -->
<!ELEMENT Module (#PCDATA)>
<!-- Which part of Service Activator generated this entry. Valid values are
FRAMEWORK (i.e. Workflow Manager or Resource Manager), or COMPONENT (i.e. a
workflow node, a plug-in, a dynamic module). -->
<!ELEMENT Part (#PCDATA)>
<!-- If the field "Part" is FRAMEWORK, Component is the specific module that
generated the entry. If field "Part" is COMPONENT, this field will indicate the
name of the component itself (i.e. the name of the workflow node or plug-in). -
->
<!ELEMENT Component (#PCDATA)>
<!-- Topic of the message. Valid values are STARTUP, RECOVERY, STATISTICS, and
COMMON_OPERATION. -->
<!ELEMENT Topic (#PCDATA)>
<!-- The name of the thread that ran this component -->
<!ELEMENT Thread (#PCDATA)>
<!-- Unique identification number for the entry (i.e. workflow id, XID). -->
<!ELEMENT ID (#PCDATA)>
<!-- The message itself. -->
<!ELEMENT Message (#PCDATA)>
```

Example C-2   **Typical HP Service Activator Log Entry**

```
<LogEntry level="INFORMATIVE" machine="activator/15.2.114.138">
   <Time>Wed Jul 11 17:08:03 MDT 2001</Time>
   <Module>mwfm</Module>
   <Part>FRAMEWORK</Part>
   <Component>LOGGER_THREAD</Component>
   <Topic>STARTUP</Topic>
   <Thread>main</Thread>
   <Message>logger thread properly started (level 2).</Message>
</LogEntry>
```

Table C-2   **JBoss Log Files**

File	Description
boot.log	Contains log output from the initial stages of JBoss startup (prior to the deployment of applications in the directory *$JBOSS_DEPLOY*). An error in this log file indicates a problem with your JBoss installation or configuration.
server.log.*	Contains log output from the subsequent stages of JBoss startup specifically related to the deployment of applications in the *$JBOSS_DEPLOY* directory, including the Service Activator J2EE components. An error in this file can indicate either a JBoss-specific problem or a configuration problem with the Service Activator components deployed in *$JBOSS_DEPLOY*. If there is an error in this file, verify that any changes made to the Service Activator configuration files in *$JBOSS_DEPLOY* (for example, web.xml and hpovact-oracle-ds.xml) are correct.

Appendix D   Security Considerations

Attention to security issues is especially important when you deploy Service Activator in a production environment. The following checklist will help you to ensure that you have configured Service Activator properly to address security considerations.

- Make sure that the Service Activator server is a secure system. To accomplish this, we recommend you obtain a security hardening/lockdown tool that can identify potential vulnerabilities. Listed below, are a few that you should consider:

  — HP-UX 11i v1 and 11i v2: HP-UX Bastille (part number B6849AA) for hardening and lockdown. Security Patch Check (B6834AA) to alert you to security-related patches that you should install.

  — Solaris 10: Solaris Security Toolkit (JASS) for system hardening.

  — Windows 2003: Microsoft Baseline Security Analyzer for system hardening.

- Set up a secure link between the Service Activator server and the target machines. This is especially important because potentially sensitive information (such as user name and password of an account created on the target machine) is transmitted to the target machine. Only authorized users can perform service activations, so it is important to keep user names and passwords secure.

  To accomplish this, Service Activator employs Secure Shell technology. Instructions for installing and configuring Secure Shell are included in the HP Service Activator—Installation Guide located on the Service Activator installation compact disc.

- Set up a secure link between web browsers and the Operator UI. When an operator logs in to the Operator UI, he or she passes user name and password information through the web browser. This communication path must be secure so that the information is not intercepted. See the HP Service Activator—Installation Guide for information about securing communication between Apache Tomcat and your web browser.

- Configure the Workflow Manager to perform authentication and authorization. The default configuration of the Workflow Manager disables authorization and authentication checking of users who log in to the Operator UI. Service Activator provides an operating system authentication module that you can use to enable authorization and authentication.

  In addition, Service Activator supports the concept of roles to control which operations a user can perform. Enable the operating system authentication module, and set up roles to control access to various capabilities. See HP Service Activator—Workflows and the Workflow Manager for more information.

- Provide secure communication between the OSS environment and the Workflow Manager. Service activation requests passed from the OSS environment to the Workflow Manager may need to be secure because customer-sensitive data (such as user names and passwords) can be passed as part of the activation request. In addition, the communications may need to be secure to ensure that unauthorized agents do not make activation requests. You can address these security requirements by deploying SSL as the communications transport between the OSS environment and the Workflow Manager. We provide details on how to modify the socket listener module in the Workflow Manager to exploit SSL in HP Service Activator—Workflows and the Workflow Manager.

- Configure Service Activator to require authorization and authentication when deploying PARs and compound tasks from Service Builder. The default configuration disables authorization and authentication for deploying PARs and compound tasks. One should note that communications between Service Builder and the Service Activator server cannot be encrypted. If you are running Service Builder in an insecure environment, we strongly recommend you only deploy from an instance of Service Builder residing on the Service Activator server, since authentication information is passed during the deploy step. See "Configuring Authentication or Authorization" in HP Service Activator—Developing Plug-Ins and Compound Tasks for more information about enabling this security.

- The default installations of Apache Tomcat and JBoss have all functionality enabled. If these are potential security issues in the customer environment, we recommend you manually configure both Tomcat and JBoss to allow for startup of only selected functionality.

- If you are going to operate Service Builder on HP-UX or Solaris via a remote X11 session, you might want to consider using X11 connection forwarding via Secure Shell. This is especially important if you are going to operate Service Activator in an insecure environment. Using X11 forwarding will ensure that authentication information you provide cannot be intercepted. Please consult the following site for instructions for setting up X11 connection forwarding:

  `http://www.itso.iu.edu/howto/ssh2.epl#x11`

- Service Activator uses an Oracle database as an inventory and service repository. Please make sure that you change the Oracle password from its default value to avoid its being compromised.

- Service Activator emits a number of log files containing information about the progress of specific service activations. These log entries can contain information that is specific to an individual customer. Thus, it is extremely important that these files remain secure. The best way to ensure this is to perform system hardening as mentioned earlier.

- Service Activator uses Secure Shell to communicate with target machines. Typically, a firewall will exist between the Service Activator server and the target machines. You will need to allow Secure Shell communications through the firewall. This is the only firewall hole that Service Activator requires.

- When deploying Service Activator in the customer's environment, you need to take into account the fact that the target machines on which Service Activator is operating may be in a different trust domain than the Service Activator server. When Service Activator invokes scripts on the target host, the possibility exists that the target could be hostile and attempt an attack on the Service Activator server based on the information that is returned via the execution of the script. There are two potential areas of concern that you should take into account when developing a solution for the customer:

  — A false return code could be returned by the script in an attempt to affect the branching of the workflow. Normally, this will only affect the activation activity on the target machine in question. If, however, you constructed a workflow to orchestrate activation activities on target machines in two different trust domains, a false return code generated by a script running on a target in one domain could impact the activities occurring on the other target system if the workflow logic does different things depending on the return code. Depending on the logic in the workflow, you could damage a server based on performing an inappropriate activation activity. Thus, do not construct workflows that will coordinate activation activities that span trust domains.

  — Similarly, you cannot necessarily trust the contents of stdout and stderr information returned by a script. A variety of attacks could be launched if malicious stdout/stderr information is provided to other applications involved in the activation request. Exercise caution when using stdout or stderr information. Detainting (for example, removing escape sequences like a semicolon that could cause the information to be interpreted incorrectly) may be necessary.

- Service Activator does not redeploy scripts to target machines every time they get executed. However, Service Activator restricts access to the scripts so that they cannot be modified. DO NOT modify the access permissions on scripts that have been previously deployed to the target hosts.

- Be careful handling potentially sensitive information to make sure it is not passed as an argument to a script that runs on the target machine. This information can be easily retrieved by getting a list of running processes (for example, "ps -ef" on UNIX). To avoid this problem, take the following steps:

  — Do not pass sensitive information to the script as an argument on the command line. Instead, place sensitive information in a file on the Service Activator server, copy it over to the target machine, read the file contents from the script, and then remove the file.

  — Make sure you do not consequently pass this sensitive information as an argument to other applications.

WARNING	**You are strongly urged to avoid using any application that requires you to pass sensitive information to it via command line arguments.**

Appendix E   Data Collector

In this appendix, you can find information about Data Collector, which is a tool used to automate collection of data necessary for resolving Service Activator problems, which require expert investigation and cannot be solved interactively. The tools gathers data that can be forwarded to support stuff for analysis.

Data Collector is installed automatically as part of Service Activator and does not require any additional configuration. Data Collector is a command line tool without any graphical user interface.

NOTE	Run the dc.bat script on Windows. UNIX users run the dc.sh script.

Start Data Collector

Here are the steps to start Data Collector.

1.   Start your command line interface.

2.   Change the directory to *$ACTIVATOR_BIN*

3.   Enter dc and press Return. If you use Cygwin or UNIX, enter ./dc.bat or ./dc.sh.

These are the minimum instructions for running Data Collector. In this case, it runs with the default attributes and puts the output file and the results archive (data_collector.xml and data_collector.xml.jar) in *$ACTIVATOR_VAR*/dc.

You would then send the data_collector.xml.jar file to your support representative for further analysis.

In file data_collector.xml, you can view what information was collected and where it was obtained from. Open the results archive to see the actual files collected.

For example, when you run Data Collector with the default attributes, the results archive contains these files and folders: data_collector.xml, activator.dir.var, activator.dir.etc, and jboss.dir.

If you run Data Collector with the default attributes a number of times, it overwrites the output file and the results archive each time. You can avoid this if you instruct Data Collector to save the output file and the results archive under different names or in other location than the default one. See Data Collector Command Line Options for the details about controlling Data Collector output.

Data Collector Command Line Options

In the section above, the instructions are given on how to run Data Collector with the default attributes. You can, however, control which data the tool collects, where it places the result and how the output file and the results archive are titled. Use the three command line options listed in Table E-1 to control Data Collector.

Table E-1   **Data Collector Command Line Options**

Option	Description
-t	This option determines which data is gathered by Data Collector. You provide the name of a task file when you use this attribute. See "Data Collector Task Files" below for the available task files.
-d	Defines the name of the directory to store the results archive. The results archive created by Data Collector is placed in a particular directory; then you forward the archive to HP. You can indicate any directory (even non-existent) to save the result in; Data Collector creates new directories automatically. It is important to remember though that only the results archive (data_collector.xml.jar) is saved in the user selected directory. The output file (data_collector.xml) is invariably saved in the default directory ($ACTIVATOR_VAR/dc). A copy of this file is also included in the results archive.
-x	Defines the fully qualified name of the output file generated by Data Collector (data_collector.xml, by default). The output file is an XML document, which summarizes what data and where from was collected. The results archive is automatically named after the output file, e.g. data_collector.xml.jar

Data Collector Task Files

Data Collector comes complete with 4 standard task files. When used in a command, these files control what data is collected. You can find the task files listed in Table E-2. See also "Data Collector Structure" below for where you can locate the standard task files

Table E-2   **Data Collector Task Files**

Task File	Description
default.task.xml	In addition to details about the Workflow Manager, Resource Manager, and Jboss, this task file collects the java version and the operating system name. If you enter a command without any task file name, Data Collector assumes that the task file name is default.task.xml
mwfm.task.xml	Use this task file when you want to collect details only about the Workflow Manager.
resmgr.task.xml	Use this task file when you want to collect data only about the Resource Manager.
jboss.task.xml	Use this task file when you want to collect data only about Jboss.

Using Data Collector

Now that you are familiar with starting Data Collector and the command line options, which control its work, it may be useful to review the following example of how the tool is used in Windows.

Let us assume that you need to collect data about the Workflow Manager. You want to save the results archive in your folder C:\my_data_collector. This is a new folder, which will be created by Data Collector as it processes your command.

You also want the output file and the results archive to be titled `my_data_collector_result`.

First of all, start your command line interface and change the directory:

```
cd C:/HP/OpenView/ServiceActivator/bin
```

Then, enter the command with the options to control the result to start collecting data:

```
dc -t mwfm.task.xml -d C:\my_data_collector -x my_data_collector_result.xml
```

After Data Collector completes the task, it returns the message to your command line interface window that data collection completed. You can then enter new commands. Remember to start your commands with dc.

You can now open the `my_data_collector` directory on your C: drive to view the results archive, which is titled `my_data_collector_result.xml.jar`. Note that the output file containing the details about data collected is located in the default directory, i.e. `C:\HP\OpenView\ServiceActivator\var\dc`. In this example, it is titled `my_data_collector_result.xml`.

Data Collector Exceptions

Data Collector is a command line tool; if it encounters exceptions when processing commands, it displays those exceptions in the command line interface window.

Here is the list of possible exceptions.

- Task file is incorrect (errors in XML structure)

- Context file (see below for details) is incorrect (errors in XML structure)

- Task file and context file schema version or source name are not equal

- Tool version specified by context file is incorrect. Current tool version is 1.0

- Exception collecting data (result directory cannot be created, file cannot be copied, result output directory cannot be created, the results archive cannot be closed)

Errors related to collecting data are recorded in the output file (`data_collector.xml`). Examine the content of this file to identify possible problems. The output file is generated even if data collecting errors prevent the tool from creating the results archive.

Data Collector Structure

Data Collector requires two types of files:

- task files

- context files

You can find and view the standard files shipped with Service Activator in the following two directories:

- the task files are located in `$ACTIVATOR_ETC/dc_tasks`

- the context files are located in `$ACTIVATOR_ETC/config/dc`

When entering commands, you provide a task file name. Then, Data Collector looks in the contents of the task file to determine the context file which will be used for gathering data. In other words, the task file is a simple XML document. It tells Data Collector which list of instructions to use to gather data. The context file (or the list of instructions) contains the precise instructions for gathering data, i.e. it specifies the directories and the file types your Data Collector must look for and pack together in the results archive.

Task Files

As it has been mentioned, a task file is a simple XML file telling Data Collector which list of instructions (or the context file) to use for a given task.

In Example E-1, you can find the content of the standard mwfm.task.xml task file. The context tag specifies that, when this task file is used in a command, Data Collector must use the mwfm.context.xml context file to gather data (here, "mwfm" is the beginning of the context file name). In this case, data related to the Workflow Manager is gathered.

NOTE   Remember that your task files must always be placed in $ACTIVATOR_ETC/dc_tasks

---

Example E-1   **Sample Task File**

```
<?xml version="1.0" standalone="yes"?>
<source source-name="OVSA" schema-version="1.0">
  <context context-name="mwfm"/>
</source>
```

You can create your own task files similar to the one in Example E-1. Use only the tags listed in Table E-3. Always begin your task file with the declaration of the XML version, e.g. `<?xml version="1.0" standalone="yes"?>` and nest the context tag(s) in the source tag.

Remember to save your task files as XML documents. Unlike for context files (discussed later on in this document), there are no special requirements for naming your task files. But you will use your task file names when entering commands.

In your task file, you can use several context tags to specify a number context files. This would allow you to run Data Collector once and gather data using several context files.

However, it may be more convenient to group contexts into several task files and so reduce the amount of data collected.

Table E-3   **Task File Tags**

Element	Attribute	Required	Description
source	source-name	Yes	This is the root tag of your task file. The source-name attribute value identifies the application. In case of Service Activator, the attribute value is OVSA. It must always be in upper case. It is important to ensure that your task and context files have "OVSA" as the value of this attribute.
	schema-version	Yes	XML version number. It must always be 1.0 Again, ensure that your task and context files have "1.0" as the value of this attribute.
context	context-name	Yes	This tag determines which context file(s) are used to collect data. The context-name attribute specifies the context file. This attribute value must not contain spaces and must not be longer than 20 characters. When processing your commands, Data Collector checks the value of the context-name attribute and based on that selects the appropriate context file, e.g. "mwfm" in `<context context-name="mwfm"/>` translates into the `mwfm.context.xml` context file name.

Context Files

In the previous section, the structure of the mwfm.task.xml task file was explained. When you use this task file in your command, Data Collector looks for the mwfm.context.xml context file to obtain instructions about what data must be gathered and where it must be gathered from. In Example E-2, you can view the content of the standard `mwfm.context.xml`.

Example E-2   **Sample Context File**

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE source-recon SYSTEM "context.dtd">
<source-recon tool-version="1.0" schema-version="1.0" source-name="OVSA">
<component name="MWFM">

   <version>
      <from-class class-name="com.hp.ov.activator.util.ActivatorVersion"
      method-name="getVersion"/>
   </version>

   <status>
      <default value="unknown"/>
   </status>

   <dir sys-dir="activator.dir.etc" dir-path="config">
      <file-filter description="MWFM configs" type="configuration"
      ascii="true"/>
   </dir>

   <dir sys-dir="activator.dir.var" dir-path="log" tree-list="true">
      <file-filter prefix="mwfm" description="MWFM logs" type="log"
      ascii="true"/>
   </dir>

   <check-sum sys-dir="activator.dir" dir-path="lib" include-dirs="false"/>

</component>
</source-recon>
```

Let us review this file in greater detail and see how you can create your own context files.

Just like task files, context files are XML documents. It is important to mention that strict naming requirements apply to context files. Context file names must always end with `.context.xml`. The beginning of a context file name must always coincide with the value of the context-name attribute of the <context> tag in the corresponding task file, e.g. `<context context-name="mwfm">` in the `mwfm.task.xml` task file directly refers to the `mwfm.context.xml` context file.

On the first line of your context file, you type an XML declaration `<?xml version="1.0" standalone="no"?>`.

Context files have a strictly defined structure, which is controlled by the `context.dtd` file in `$ACTIVATOR_ETC`/config/dc. Only the elements which are defined in `context.dtd` file can be used in context files. The occurrence of those elements and their attributes are dependent on the context.dtd file as well. Make sure to include context.dtd in your DOCTYPE reference `<!DOCTYPE source-recon SYSTEM "context.dtd">`.

Next, add the root tag `<source-recon tool-version="1.0" schema-version="1.0" source-name="OVSA">`, which will contain all other tags in your context file. The tag attributes are required. You can find their detailed descriptions in Table E-4.

Open the <component> tag and add the name attribute <component name="MWFM">. The value of this attribute indicates what data is gathered (in this case, data related to the Workflow Manager).

Open the <version> tag. Then, add the <from-class> tag. `<from-class class-name="com.hp.ov.activator.util.ActivatorVersion" method-`

name="getVersion"/> tells Data Collector where the version number of the present component can be found. Close the <version> tag.

Open the <status> tag. In Example E-2, the pair of <status> tags contain <default value="unknown"/> This indicates that Data Collector will not try to determine whether the component is installed or not. See Table E-4 for other options. Close the <status> tag.

Open the <dir> tag. Its attributes specify the directory in which Data Collector must look for files. For example, <dir sys-dir="activator.dir.etc" dir-path="config"> directs Data Collector to ...\ServiceActivator\etc\config. Such directory names as <activator.dir.etc> are specified in the parameters of Data Collector.

NOTE	As its value, the sys-dir attribute can have one of the following directory names: activator.dir, activator.dir.etc, activator.dir.var, jboss.dir, other.dir. If you set the attribute value to other.dir, Data Collector will not attempt to construct the directory name from the values of the sys-dir and the dir-path attributes. Instead, it will expect to find the full directory name in the dir-path attribute. Such functionality is intended for those cases when you need to write a context file for gathering data in the directories which are not covered by the other four standard directory names.

After you have defined the opening <dir> tag and its attributes, add the <file-filter> tag and (optionally) the <data-file> tag. These two tags can only appear inside a pair of the <dir> tags. According to their attributes, Data Collector filters and gathers files and folders in the directory specified by the attributes of the <dir> tag. For example, <file-filter prefix="mwfm" description="MWFM logs" type="log" ascii="true"/> limits the selection to the log files that have "mwfm" in their names (prefix="mwfm" type="log"), and are encoded in ASCII (ascii="true"). See Table E-4 for other attributes that control how files are filtered. After you have finished defining the result filtering options, close the <dir> tag.

You can go on and add more <dir> tags, which specify other directories where Data Collector must gather information from. Remember to add filtering options.

After you have defined your <dir> tags, add the <check-sum> tag. According to this tag Data Collector forms a list of files.

Finally, add the closing </component> tag and the closing </source-recon> tag. Save you context file as an XML file. Remember to follow the obligatory naming requirement (context file names must always end with .context.xml).

NOTE	Remember that your context files must always be placed in $ACTIVATOR_ETC/config/dc

Table E-4   **Context File Elements**

Element	Attribute	Required	Element Content	Description
source-recon			(java-version, runtime-platform)?, licence?, component*, checksum	This is the root tag of a context document.
	tool-version	Yes		Specifies the version of Data Collector
	schema-version	Yes		Specifies the schema version of context XML file
	source-name	Yes		Unique identifier of the application, e.g. "OVSA" stands for Service Activator
java-version				Java version of current JVM
runtime-platform				Name of the run-time platform
component			(version, status, dir*, checksum*)	This element allows you to get information about a specific system component, e.g. the Workflow Manager
	name	Yes		Specifies the name of a system component

Element	Attribute	Required	Element Content	Description
check-sum				This element allows you to get a list of files and directories in a specified directory. The name check-sum is somewhat misleading because it actually does not calculate the check-sum for the collected files and directories it only creates a list.
	sys-dir	No		Specifies the main directory, e.g. `activator.dir`, `activator.dir.etc`, `activator.dir.var`, `jboss.dir`. These directory names are specified in the parameters of Data Collector. The default value is other.dir.
	dir-path	Yes		Specifies the exact directory, e.g. if `sys-dir="activator.dir.etc"` and `dir-path="config"`, the directory is `.../ServiceActivator/etc/config`
	includedirs	No		In addition to files, lists directories (can be true or false). The default value is "true".
version			fromclass|default	Gets the version of a component. It has two options to obtain the version: (1) use from-class, (2) default. The default tag may be used when the class which may return right version using from-class tag does not exist.
status			fromclass|default	This is currently not used in Service Activator. Valid values are "installed", "not installed", "supported", "unsupported" and "unknown"

Element	Attribute	Required	Element Content	Description
dir			filefilter*, data-ile*	This element specifies the directory from which files must be collected.
	sys-dir	No		The attribute specifies the main directory, e.g.`activator.dir.` These directory names are specified in the parameters of Data Collector. The default value is `other.dir`.
	dir-path	Yes		Specifies the exact directory, e.g. if `sys-dir="activator.dir.etc"` and `dir-path="config"`, the directory is `.../ServiceActivator/etc/config`
	tree-list	No		If this attribute is set to "true", then the results of gathering data are presented in tree view. The root directory in the results directory has the value of the "sys-dir" attribute as its prefix. The attribute may be set either to "true" or to "false". The default value is "true".
from-class				Allows you to receive information form the class you specify.
	class-name	Yes		Class name from which information must be gathered.
	method-list	Yes		Specified method must not have any parameters and return values as String.
default				This tag allows you to set the default value.
	value	Yes		Default value

Element	Attribute	Required	Element Content	Description
file-filter				The file filter element allows you to filter elements from a list. This element may be used only inside of element "dir". Element "dir" specifies the directory, so this element should filter files only inside this directory
	prefix	Implied		Specifies the prefix for filtered files
	suffix	Implied		Specifies the suffix for filtered files
	newest	Implied		Specifies the number of the newest files which must be gathered.
	description	Yes		Short description of files
	type	No		Files type must be one of: configuration; log; trace; data. The default value is "configuration".
	ascii	No		Tells whether the file is ASCII or not (values may be "true" or "false"). The default value is "true".
	include-sub-folders	No		Gather files from sub-folders of specified location (values may be "true" or "false"). The default value is "false".

Element	Attribute	Required	Element Content	Description
data-file				File which is needed to copy to result directory of Data Collector. This element may be used only inside of the "dir" element. The "dir" element specifies the directory, so this element should specify only the file name (attribute "name") (etc. "config.xml", but not "/etc/config.xml").
	name	Yes		File name
	description	Yes		Short description of file
	type	No		File type must be one of: configuration; log; trace; data. The default value is "data".
	ascii	No		Indicates if the file is ASCII or not (values can be "true" or "false"). The default value is "false".
licence			from-class | default	Gets the license information. It has two options to obtain the license information: (1) use from-class, (2) - default. The default tag may be used when the class which may return right version using from-class tag does not exist

Review

- Data Collector is an independent part of Service Activator. It is used for gathering data requested by support in those cases when issues appear in Service Activator.

- Data Collector is a command line tool and does not have any graphical user interface. Use your command line interface to start Data Collector.

- You can control what results Data Collector generates by using these command line options: -t -d -x

- Data Collector generates the data_collector.xml output file. Review content of this file to see what data was gathered and where it was obtained from. The files gathered are packed in the data_collector.xml.jar archive. By default, Data Collector puts the output file and the results archive in $ACTIVATOR_VAR/dc

- To gather data, Data Collector relies on task files and context files. Use task files to refer Data Collector to context files, which are lists of instructions for gathering data.

- Both, task and context files are simple XML documents. If you decide to create your own context files, remember to place them in the correct directory. Otherwise, Data Collector will not be able to access them. You can place task file in a directory of your choice. However, you will have to provide path to that directory when you enter commands. It may be more convenient to place task files in the default directory, in which case, you will have to provide only task file names in your commands.

- Elements of context files are controlled by `context.dtd`. If you decide to create your own context files, review the content of `context.dtd` to see the valid elements and the rules applicable to their occurrence in context files. Also, remember the naming requirement when you save your context files.