HP OpenView Smart Plug-In for Remedy Action Request System integration:

Administrator's Reference

HP OpenView

Version A.02.50

October 2004

Legal Notices

Contents

Contents

Contents

Contents

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: April 2000

Second Edition: December 2001

Third Edition: September 2002

Fourth Edition: January 2003

Fifth Edition: August 2003

Sixth Edition: June 2004

Seventh Edition: August 2004

Eighth Edition: October 2004

Conventions

The following typographical conventions are used in this manual.

Table 1   Typographical Conventions

Font Type	What the Font Type Represents	Example
Italic	Book or manual titles, and man page names	Refer to the *HP OpenView Smart Plug-In for Remedy Action Request System integration: Administrator's Reference* and the *opc(1M)* manpage for more information.
	Provides emphasis	You *must* follow these steps.
	Specifies a variable that you must supply when entering a command	At the prompt type: rlogin *your_name* where you supply your login name.
	Parameters to a function	The *oper_name* parameter returns an integer response.
Bold	New terms	The **monitor agent** observes...
Computer	Text and items on the computer screen	The system replies: Press Enter
	Command names	Use the grep command ...
	Function names	Use the opc_connect() function to connect ...
	File and directory names	/opt/OV/bin/OpC/
	Process names	Check to see if opcmona is running.
	Window/dialog box names	In the Add Logfile window...
Computer Bold	Text that you must enter	At the prompt, type: **ls -l**
Keycap	Keyboard keys	Press **Return**.
[Button]	Buttons on the user interface.	Click [Operator]. Click the [Apply] button.

Table 1   **Typographical Conventions**

Font Type	What the Font Type Represents	Example
`Menu Items`	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select `Actions:Utilities-> Reports...`

The OV SPI for Remedy Documentation Map

This section illustrates what information is available for the HP OpenView Smart Plug-In for Remedy Action Request System integration (OV SPI for Remedy) and where you can find it. The following manuals are available in with the HP OpenView Smart Plug-In for Remedy Action Request System integration:

- *HP OpenView Smart Plug-In for Remedy Action Request System integration: Administrator's Reference*

- *HP OpenView Smart Plug-In for Remedy Action Request System integration: Software Release Notes*

Note that only the PDF format is available in the `SPI-RemedyDoc` software bundle. For more information on the OV SPI for Remedy software bundles, see Table 2-4 on page 45.

In addition to the documentation for the HP OpenView Smart Plug-In for Remedy Action Request System integration, related OpenView products also provide a comprehensive set of manuals, which aim to assist you in using the products and improving your understanding of the underlying concepts.

Table 2   **Documentation Roadmap**

HP OpenView Product Name	Manual Title
OpenView Operations for Unix	Concepts Guide Installation Guide for the Management Server Administrator's Reference (Volumes 1 &2) Error Message Reference Software Release Notes

1 Introducing the OV SPI for Remedy

This section introduces the **HP OpenView Smart Plug-In for Remedy Action Request System integration** software. The section covers the following areas:

- Overview of the HP OpenView Smart Plug-In for Remedy Action Request System integration

- Understanding OVO/Unix

- Understanding the AR System

- The Functionality of the OV SPI for Remedy

- The OV SPI for Remedy Architecture

Introducing the OV SPI for Remedy
**Overview of the HP OpenView Smart Plug-In for Remedy Action
Request System integration**

Overview of the HP OpenView Smart Plug-In for Remedy Action Request System integration

The HP OpenView Smart Plug-In for Remedy Action Request System integration (OV SPI for Remedy) enables you to integrate OpenView Operations for Unix (OVO/Unix) with the Remedy Action Request System ™ (AR).

Figure 1-1   **Integration between OVO/Unix and an AR System**
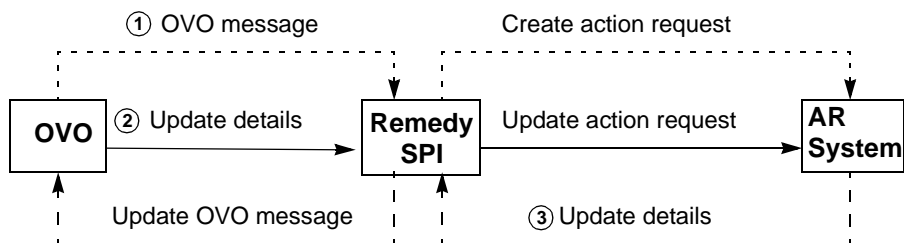
Figure 1-1 gives an overview of how the OV SPI for Remedy integrates OVO/Unix and an AR System. It shows that:

1. the OV SPI for Remedy accepts messages from OVO/Unix, then forwards the OVO/Unix message to the AR System. This creates a new action request which corresponds to the OVO/Unix message.

2. if a OVO/Unix message is modified, the OV SPI for Remedy updates the action request.

3. If an action request is modified, the OV SPI for Remedy updates the OVO/Unix message details.

For example, if OVO/Unix receives a message about a printer failure, the OV SPI for Remedy uses this message to create a new action request in the AR System. A help desk user sees the new action request and assigns it to an engineer. The engineer adds information to the action request, and the OV SPI for Remedy changes the OVO/Unix message as appropriate. After the engineer repairs the printer, they update the status of the action request. The OV SPI for Remedy then changes the

Introducing the OV SPI for Remedy
**Overview of the HP OpenView Smart Plug-In for Remedy Action
Request System integration**

status of the message in OVO/Unix.

The rest of this chapter describes the OV SPI for Remedy's functionality
and architecture in more detail.

Understanding OVO/Unix

This section gives a brief overview of the OVO/Unix features that are important for the OV SPI for Remedy. For more detailed information about OVO/Unix, see the *HP OpenView Operations Concepts Guide*.

Figure 1-2   **Overview of OVO/Unix**
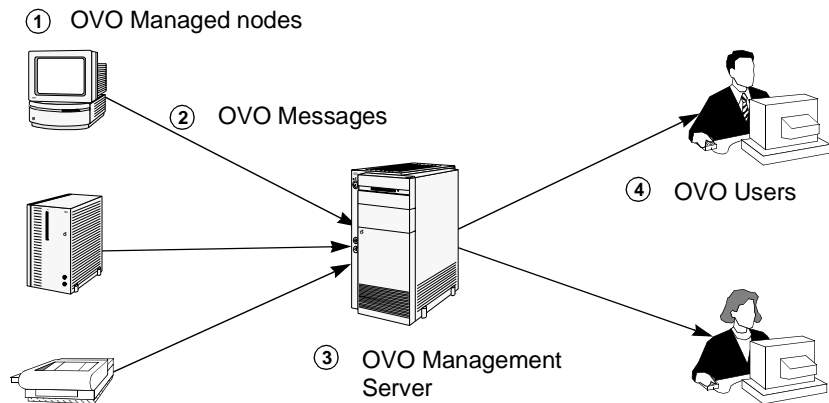
Messages

Figure 1-2 gives an overview of OVO/Unix. It shows that:

1. OVO/Unix can monitor and control remote systems, known as managed nodes.

2. If a particular status or event occurs on a managed node, the OVO/Unix agent sends a message to a management server. The messages contain structured information about the managed node and the problem that has been identified.

3. The management server receives messages from managed nodes.

4. Users can see messages for message groups and managed nodes that they are responsible for.

For example, an automatic backup fails because of a disk error. This generates an error message for the message group "Backup". The OVO/Unix agent on the managed node sends to the management server.

The management server receives the message. Users with access to the both the `Backup` message group and the managed node can then see details of the backup failure.

Message Management

When users can see a message, OVO/Unix enables them to work on it in a number of different ways. Users can:

- Take *ownership* of a message. This shows that the user is aware of the message and intends to deal with it personally. Taking ownership of a message prevents other users working on the message.

- Add *annotations* to a message. The user can attach notes to a message that give helpful and appropriate information. For example, a message annotation could describe how a user fixed the problem to which the message related so that identical or similar messages may be dealt with more efficiently in the future.

- *Escalate* a message. Users who are unable to deal with a message can send it to a different OVO/Unix management server where the necessary expertise is available.

- *Acknowledge* a message. The user can move a message into the history database. This usually happens when work on this message is complete.

Actions

Figure 1-2 shows how messages enable OVO/Unix to monitor managed nodes. At each stage, an action can be started in response to the information in the message. An action can be started on:

- the managed node where the message originated as soon as the problem is discovered (automatic action)

- the management server when a message arrives (automatic action)

- the user when they see the message (operator-initiated action)

An action can be a command, a program, a shell script, or any other appropriate response to the problem to which the message relates. The action usually attempts to solve the problem to which the message relates or provide more details about it.

For example, when the management server receives a message about a

backup failure, the management server may start an automatic action that performs the backup on a different disk. Additionally, when the user sees the message, they may identify the cause of the failure and start an operator-initiated action to correct it.

Understanding the AR System

This section gives a brief overview of the AR System features that are important in the context of the this OV SPI for Remedy. The sections look at the following areas:

- The ARS Work Flow
- The AR Form

The ARS Work Flow

The Remedy Action Request System allows users to work on a database containing so-called **Action Requests—Trouble Tickets** in OVO/Unix terminology. Typically, these Action Requests (AR) are created either by users who encounter a problem during the performance of their every-day work or by first-line support people, who receive reports of problems by means of other channels such as phone or e-mail.

An AR is essentially a description of a problem. The problem description contains all available information such as; problem category, affected components, the type of problem, a serial number, the people the problem affects, the person who submitted the action request, and so on. One of the major features of ARS is the flexibility the user is allowed in the design of the structure of the AR. The structure of an AR is defined in so-called **form** (schema in previous versions). A form consists of a set of fields, which hold the individual pieces of information which, brought together, provide a comprehensive description of a problem. Clearly, each AR created from the same form contains the same set of fields. However, a user can adapt a form to suit the demands of a particular environment by defining the set of fields needed to hold all the necessary information.
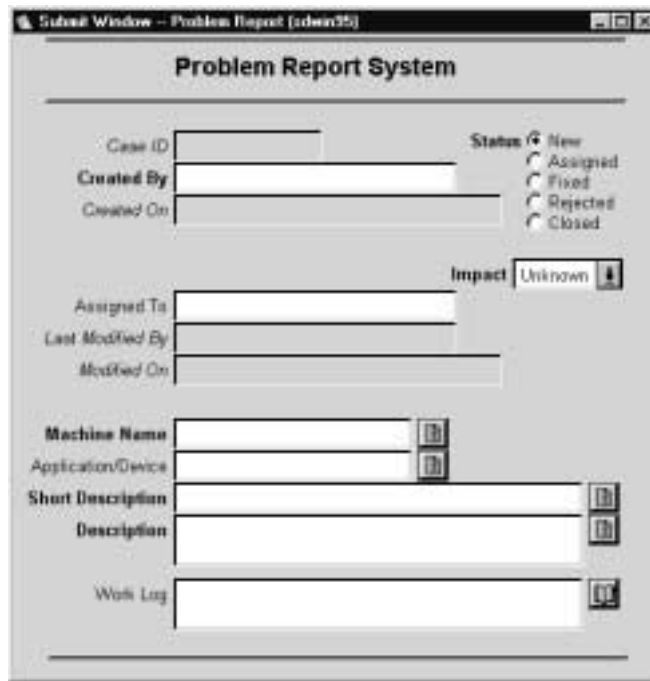
Once an AR has been created, it resides in the AR System database as a **problem report**. As soon as the problem itself has been solved, for example by an external action, the corresponding problem report can be deleted. However, you can choose to keep for future reference all the additional notes and general information which were added by the support people who worked on the problem—in case the same, or a similar, problem occurs again.

The AR Form

The AR System enables users to work on a database of action requests. Each action request usually contains details of a problem that needs attention. For example, an action request may contain details of a printer failure.

Figure 1-3   **A Simple AR Schema**

Figure 1-3 shows a simple AR System form. A form consists of fields that can contain single items of information. Together, the fields in a form define the structure of one type of action request.

The AR System enables an organization to design forms that meet its own needs. The organization is responsible for deciding what forms they need and what fields the forms have. Important for this OV SPI for Remedy is that:

- each form has a unique name.

- each field within a form has a unique identifier.

- each field within a form has a distinct data type, for example:

  — character

  — diary

  — date/time

If forms exist, they enable users to create, view, and modify action requests. For example, if an organization's help desk uses the AR System and someone calls the help desk to report a problem, the user can enter details into the appropriate form. The act of entering the details creates a new action request. After someone solves the problem, the form may be used to update the status of the action request, for example: to add details of the cause of the problem and any helpful hints about how the problem was solved.

AR Filters

A **filter** is defined in ARS as any action or group of actions carried out by ARS on the AR server system. As requests are made to the database, filters are applied, for example, to check values in the fields of forms that have been completed and submitted. You can configure the automatic trigger of actions depending on the result of the filter.

The OV SPI for Remedy uses the AR filter mechanism to trigger communication between the ARS and HP OpenView OpenView Operations for Unix. The OV SPI for Remedy client, which resides on the AR server is registered as a filter and configured to react to events such as *modify* and *delete*. In this way, changes to AR forms that are of interest to OVO/Unix can trigger automatic updates in the OVO/Unix database via the OV SPI for Remedy. The OV SPI for Remedy keeps a constant watch on these changes and ensures that the two databases are kept in constant synchronization. For more information on filters, see "ARS Filters and Active Links" on page 146.

AR Active Links

An **active link** is defined in ARS as any action or group of actions carried out by ARS on the AR client—in simple terms, the user interface. Active links are typically triggered by user actions, for example the click of a mouse button in the GUI.

The OV SPI for Remedy uses this feature to provide buttons in AR forms which allow the user to launch operator-initiated actions in OVO/Unix from within ARS or even escalate OVO/Unix messages so that the

necessary expertise can be obtained quickly in order to fix a problem. For more information on active links, see "ARS Filters and Active Links" on page 146.

The Functionality of the OV SPI for Remedy

The OV SPI for Remedy enables you to integrate OVO/Unix's monitoring and control capabilities with an AR System. When OVO/Unix detects problems in the environment, the OV SPI for Remedy can immediately and automatically create an action request even before the first human notices the situation. In addition, the OV SPI for Remedy maintains consistency between action requests and OVO/Unix messages until the problem is solved.

Key Features

The OV SPI for Remedy can:

- adapt to fit an organization's individual needs

- create new action requests by automatically sending a OVO/Unix message to an AR System

- create new action requests when a OVO/Unix user decides to send one or more OVO/Unix messages to an AR System

- use external data and actions to add supplementary details to an action request

- update action requests when the corresponding OVO/Unix message changes

- update action requests by adding further OVO/Unix messages

- update OVO/Unix messages from the AR System. This includes:

    — changing ownership of the OVO/Unix message
    — adding annotations to the OVO/Unix message
    — escalating a OVO/Unix message
    — acknowledging a OVO/Unix message to reflect the status of the action request
    — starting an action in OVO/Unix

The OV SPI for Remedy also provides:

- complete integration with the OVO/Unix GUI

- the capacity to monitor AR system availability with OVO/Unix

- the capacity to monitor OV SPI for Remedy availability with

OVO/Unix

Example Scenarios

The following two scenarios are examples which describe how you could use the OV SPI for Remedy in your environment. Of course, you can use the OV SPI for Remedy in many other ways, too, depending on the needs of you organization.

Figure 1-4   **An Action Request that Provides Information**
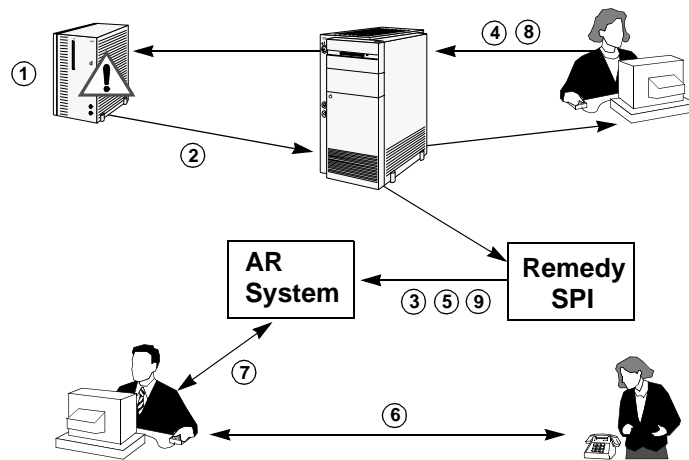
Figure 1-4 shows an example of how the OV SPI for Remedy can create and update an action request to provide information.

1. A backup procedure fails because the disk is full.

2. The OVO/Unix agent on the managed node sends a message to the OVO/Unix management server.

3. The OVO/Unix management server automatically tells the OV SPI for Remedy to create a new action request via the OVO/Unix Trouble-Ticket interface.

4. The OVO/Unix user sees the message and decides to start the backup on a different disk. The user explains what he is doing in the annotation he adds to the OVO/Unix message.

5. The OV SPI for Remedy automatically updates the action request.

6. An end-user calls the help desk because he cannot find the backup of an important file.

7. The help desk staff see that an action request already exists for this problem. They read the information in the Diary field and tell the caller that the backup is on a different disk.

8. Meanwhile, the OVO/Unix user solves the problem of the full disk and acknowledges the OVO/Unix message.

9. The OV SPI for Remedy automatically updates the information associated with the action request and the action request's status.

Figure 1-5　　　**An Action Request that Needs Attention**
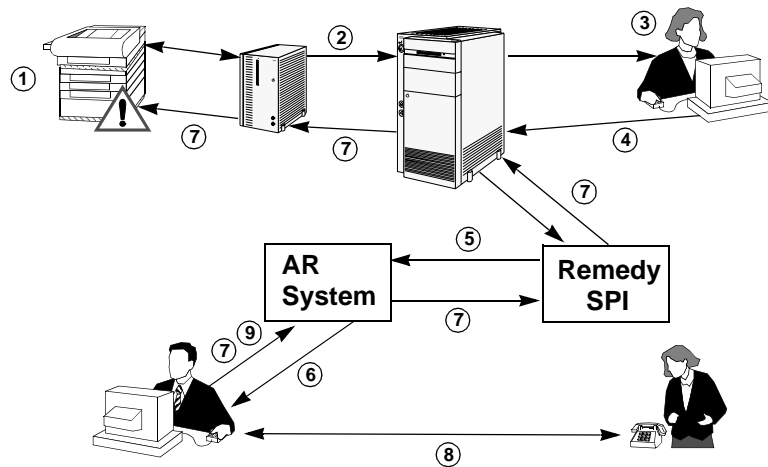
Figure 1-5 shows an example of how the OV SPI for Remedy can create an action request that needs attention, then update the OVO/Unix message when the action request changes.

1. A large printer in the print room fails because of a paper jam.

2. The OVO/Unix agent on the managed node sends a message to the OVO/Unix management server at a different site.

3. The OVO/Unix operator sees the message and diagnoses the problem. The OVO/Unix operator adds an annotation to the OVO/Unix message explaining that someone needs to remove the jammed paper and then reset the printer.

4. Since the OVO/Unix operator cannot remove the jammed paper from a printer in a remote site, he decides to forward the OVO/Unix message to the AR System.

5. The OV SPI for Remedy receives the OVO/Unix message and creates a new action request.

6. The help desk staff see that an action request exists for a printer problem and that there is a suggested solution. They call a technician and ask them to remove the jammed paper from the printer.

   As soon as the Help Desk starts to deal with the action request, the OVO/Unix operator `Remedy SPI User` automatically takes ownership of the OVO/Unix message linked to the Action Request. Ownership of an OVO/Unix message indicates to other OVO/Unix operators that the problem is being solved.

7. The help desk staff start an action in OVO/Unix that resets the failed printer.

8. The technician reports that the printer is working, so the help desk staff close the action request.

9. The OV SPI for Remedy automatically acknowledges the corresponding OVO/Unix message.

The OV SPI for Remedy Architecture

This section gives a brief overview of the architecture of the OV SPI for Remedy and some of the underlying technical concepts. The section covers:

- Process and Network Components

- Data Components

- OV SPI for Remedy Components Integrated with OVO/Unix

Process and Network Components

To use the OV SPI for Remedy, the components shown in Figure 1-6 at the very least must be present, namely:

- one OVO/Unix management server with a OV SPI for Remedy server

- one AR System server with a OV SPI for Remedy client (if you want changes to the action requests to update pro-actively any associated OVO/Unix messages)

Figure 1-6   **Server and Network Architecture**
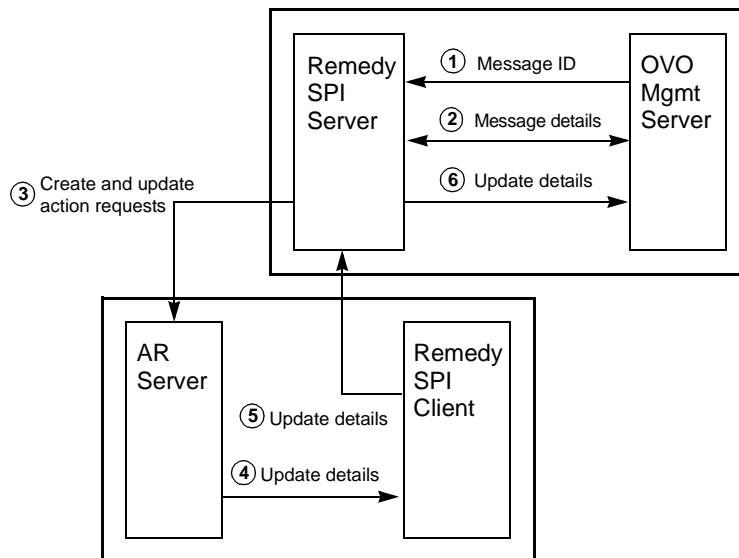
Figure 1-6 on page 31 shows that the OV SPI for Remedy server runs on the same machine as the OVO/Unix management server. The AR server can run on the same or a different machine. The OV SPI for Remedy client must be on the same machine as the AR server. Figure 1-6 also shows:

1. The OV SPI for Remedy server receives the ID of a OVO/Unix message from the OVO/Unix management server via a submission program.

2. The OV SPI for Remedy server retrieves details of the OVO/Unix message from the OVO/Unix management server via an application programmer interface (API).

3. The OV SPI for Remedy server creates and updates action requests using the ARS API.

4. When an action request changes, the AR System calls the OV SPI for Remedy client and passes on the details of the update via the ARS filter mechanism.

5. The OV SPI for Remedy client sends the update details to the OV SPI for Remedy server.

6. The OV SPI for Remedy server updates the OVO/Unix message on the OVO/Unix management server using OVO/Unix's API.

7. When a OVO/Unix message is modified, the OV SPI for Remedy server receives a message-change event via the OVO/Unix API.

8. OVO/Unix sends the details of the message-change event to the OV SPI for Remedy server via the OVO/Unix API.

9. The OV SPI for Remedy server updates the AR via the ARS API.

Figure 1-7   **Multiple Servers**
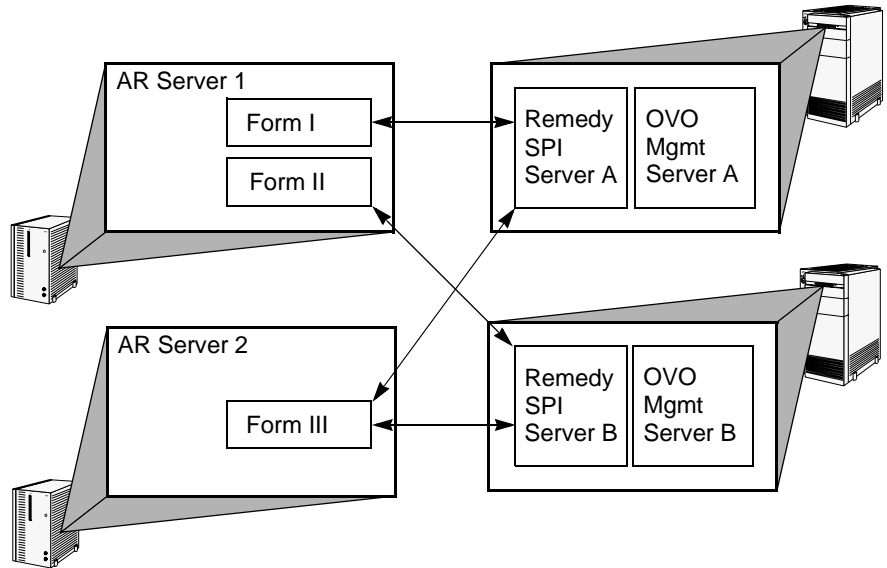
The OV SPI for Remedy also supports multiple OVO/Unix management servers and multiple AR servers. Figure 1-7 shows that each management server has one OV SPI for Remedy server. Each OV SPI for Remedy server can create and update action requests on any of the AR servers. In addition, each AR server has a OV SPI for Remedy client that can update OVO/Unix messages on any of the OVO/Unix management servers.

Data Components

This section describes what data components are activated and used when data flow between OpenView Operations for Unix and the OV SPI for Remedy. The section covers the following areas:

- External Actions and Data

- Configuration File

- Rules File

- The OV SPI for Remedy Database

Figure 1-8 shows how data flows in and out of the OV SPI for Remedy server when a OVO/Unix message is submitted to the AR system.

Figure 1-8   **Data Flow in the OV SPI for Remedy**
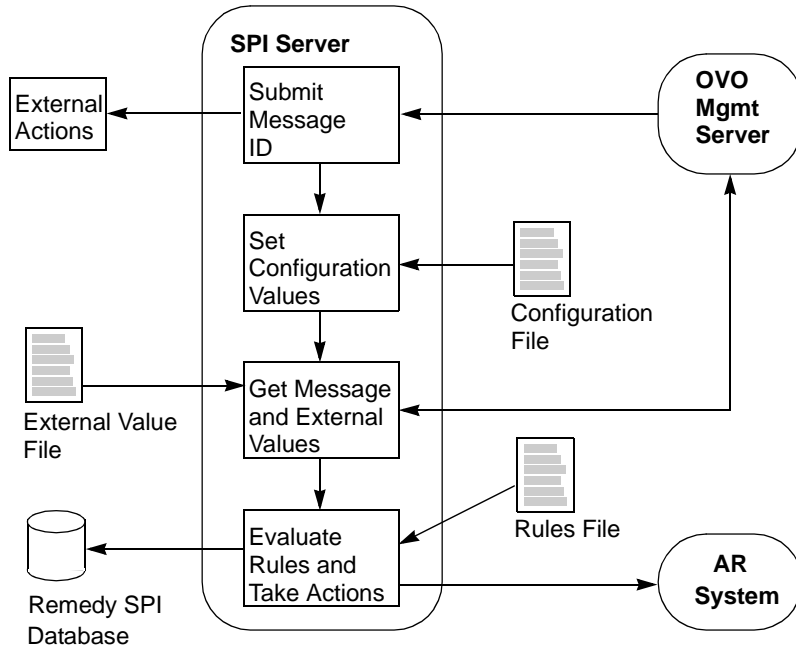
External Actions and Data

To submit a OVO/Unix message to the OV SPI for Remedy, OVO/Unix passes the message's ID to a script called remspisub_auto.sh (via the Trouble Ticket Interface) and remspisub_man.sh (via an OVO/Unix application). These scripts have a special user code section where you can add commands, for example; to start any external actions that are necessary. This could include actions to update or retrieve data in an external database.

The user code section could also include actions that generate external value files. External value files contain additional information that the OVO/Unix message does not provide. The OV SPI for Remedy can add this extra information when it creates a new action request.

After the user code section, the message ID is submitted to the OV SPI for Remedy server.

For more details about using external actions and external value files, see "Using External Actions and Data" on page 164.

Configuration File

The configuration file `remspi.cfg` defines the user name and passwords that the OV SPI for Remedy uses to log in to OVO/Unix. The passwords are encrypted. For more details about creating and modifying the configuration file, see "Configuring Communication Between OVO/Unix and ARS" on page 59.

Rules File

The rules file uses OVO/Unix message attributes and additional external values to determine:

- on which AR system an action request should be created

- which AR form (schema) to use

- which OVO/Unix message attributes match which AR form fields

The `rules` file defines *targets* and *conditions*. A target defines the *name* of a form on an AR server. The target also defines how the OV SPI for Remedy should update a OVO/Unix message if one of this schema's action requests changes.

The `rules` file also has a set of conditions. Conditions define criteria that filter OVO/Unix messages and determine what actions need to be taken to address the problems described in the message. If a OVO/Unix message meets a condition, the OV SPI for Remedy can, for example, create a new action request on the targets which are tied to that condition.

A condition also defines the fields in the schema that the OV SPI for Remedy completes to create an action request. The OV SPI for Remedy can complete fields in a schema using data from the original OVO/Unix message and also from an external value file.

You create your own rules file to define targets and conditions that meet your organization's needs. For more details, see "Customizing the Rules File" on page 95.

The OV SPI for Remedy Database

The OV SPI for Remedy maintains information in the databases relating to the associations between action requests and OVO/Unix messages. Every time the OV SPI for Remedy creates a new action request, it adds a new record containing the OVO/Unix message ID and the action request ID. There is one database for each target in the rules file,

because one OVO/Unix message may have associations with several targets.

When a OVO/Unix message changes, the OV SPI for Remedy uses the database to find the ID or IDs of associated action requests.

OV SPI for Remedy Components Integrated with OVO/Unix

The OV SPI for Remedy also provides components that integrate with OVO/Unix and the OVO/Unix GUI, including:

- node groups for:
  - — AR servers UX
  - — AR servers NT
  - — OV SPI for Remedy servers
- message groups for:
  - — AR servers
  - — OV SPI for Remedy servers
- default templates to monitor:
  - — AR server processes
  - — AR System log files
  - — the OV SPI for Remedy server process
  - — OV SPI for Remedy log files
  - — the communication queue between OVO/Unix and the SPI
- operator and administrator users
- applications to:
  - — submit a OVO/Unix message
  - — attach OVO/Unix messages to an existing action request
  - — start a OV SPI for Remedy server
  - — stop a OV SPI for Remedy server
  - — display the OV SPI for Remedy server status

— test the OV SPI for Remedy

— start an AR server

— stop an AR server

2 Installing the OV SPI for Remedy

This section describes how to install and upgrade the HP OpenView Smart Plug-In for Remedy Action Request System integration. The section covers the following areas:

- Installation Prerequisites

- Installing the OV SPI for Remedy

- Upgrading the OV SPI for Remedy Software

NOTE   The OV SPI for Remedy is available on the *HP OpenView Operations for Unix SPI CD-ROM*. The commands for installing from the different CDs are slightly different and there is a separate section for this.

Installation Prerequisites

The OV SPI for Remedy consists of client and server components and integrates with existing OVO/Unix and AR servers. The OV SPI for Remedy client runs on the same machine as the AR server. The OV SPI for Remedy server runs on the same machine as the OVO/Unix server.

Table 2-1 shows the software prerequisites for the OVO/Unix servers on HP technical workstations and business servers, as well as on Sun SPARC systems.

Table 2-1   **Supported OVO/Unix Servers**

PRODUCT VERSION	Operating System				
	HP-UX		Sun Solaris		
	11.0	**11.11**	**7**	**8**	**9**
OVO/Unix 7.1+	X	X	X	X	X
OVO/Unix 8.0 / 8.1	X	X	X	X	X

NOTE   The HP OV SPI for Remedy is linked with AR libraries 4.5.2 on all supported OVO management server platforms. For a client program using the 4.5.2 libraries, the list of possible target AR server versions is 4.0.3, 4.5.2, 5.0.1 and 5.1.*. In addition, connection to AR server versions 6.0 and 6.0.1 is possible, too.

Table 2-2 shows software prerequisites for the AR servers on the various

supported hardware platforms.

Table 2-2   **Prerequisites for the AR Server**

Hardware Platform	Operating System	AR System Version			
		4.0.3[(*)]	4.5.2	5.0.1[(+)]	5.1.* 6.0 6.0.1
HP Technical Workstations and Business Servers	HP-UX 10.20	X	-	-	-
	HP-UX 11.0	X	X	X	X
	HP-UX 11.11	X	X	X	X
Intel	Windows NT 4.0 Windows 2000	X	X	X	X
Sun SPARC	Solaris 2.6	X	X	X	X
	Solaris 7	X	X	X	X
	Solaris 8	X	X	X	X
	Solaris 9	X	X	X	X

NOTE   To take full advantage of the OV SPI for Remedy, your AR servers should be OVO/Unix managed nodes.

The prerequisites listed above do not invalidate the prerequisites for the AR server mentioned by the manufacturer (Remedy Corp.).

[(*)] Although the OV SPI for Remedy is able to connect to an AR server 4.0.3, using this version is not encouraged, since the Remedy Corp. stopped supporting it.

[(+)] Not supported. Due to a problem in ARS 5.0.1, it's not possible to start operator initiated actions or to escalate a message using the bundled AR form **Problem Report**. This problem does not occur with ARS 5.1.0 and higher.

Disk-Space Requirements

Table 2-3 lists the disk space requirements for the OV SPI for Remedy software on both the OVO/Unix management server and the AR server, which is the OV SPI for Remedy client.

Table 2-3   **Disk-Space Requirements**

Machine	OVO/Unix Version	OS	Installation (MB)	Runtime Files (MB)	Total
OVO/Unix Server	all	all	32	8	40
AR Server [a]	n/a	n/a	n/a	5	5

a. OV SPI for Remedy client

Memory (RAM) Requirements

There are no specific requirements concerning the amount of RAM installed on either the OVO/Unix or the AR server.

Installing the OV SPI for Remedy

The installation steps differ very slightly depending on whether or not you want your AR server to be a OVO/Unix managed node. The instructions in this section assume that you *do* want your AR server to be a managed node. In fact, we strongly recommend it in order for you to be able to make use of the OV SPI for Remedy's powerful monitoring capabilities, for example; to monitor the health and status of the AR server processes. If you do *not* want your AR servers to be OVO/Unix managed nodes, please pay special attention to step 5 in the instructions below.

In order to install the OV SPI for Remedy, you need to complete the steps indicated in the list below and described in more detail in the sections that follow:

1. Carry out some brief pre-installation checks

2. Install the OV SPI for Remedy software on the OVO/Unix management server

3. Verify that the installation of OV SPI for Remedy software bundle completed successfully on the OVO/Unix management server

4. Assign and distribute the appropriate OV SPI for Remedy templates to the OVO/Unix management server

5. Either:

   a. Assign and distribute the appropriate OV SPI for Remedy templates, along with all the rest of the OV SPI for Remedy software, to the AR server (the OVO/Unix managed node)

      or:

   b. If you do *not* want your AR server to be a OVO/Unix managed node, copy the `/var/opt/OV/share/databases/OpC/mgd_node/customer/<platform>/cmds/itoupdate` (itoupdat.exe for Windows) file from the OVO/Unix management server to the AR server manually.

      Ideally, the file should be copied to the location indicated in Table 2-9, "File Locations for the Installed OV SPI for Remedy Client," on page 51. However, as long as you tell ARS where to find it, the file can be placed in the location of your choice.

After you complete the installation, you need to configure the ARS filters to use the itoupdate file to communicate with OpenView Operations for Unix. For more information, see "ARS Filters and Active Links" on page 146.

Pre-Installation Checks

Before you start the installation of the HP OpenView Smart Plug-In for Remedy Action Request System integration, it is important to make sure that you are installing the corrects software files. The following tables list the bundles, products, and filesets that make up the OV SPI for Remedy and explain what the contents are. Table 2-4 explains which software depot (SD) bundles are available with the HP OpenView Smart Plug-In for Remedy Action Request System integration.

Table 2-4   **The OV SPI for Remedy SD Bundles**

SD Bundle	Product	Description
SPI-Remedy	SPI-REM-SRV	OV SPI for Remedy server software – contains everything necessary to run the OV SPI for Remedy (executables, shell scripts, etc.)
	SPI-REM-CFG	
	SPI-REM-DOC	
SPI-RemedyDoc	SPI-REM-DOC	OV SPI for Remedy documentation

Table 2-5 explains which filesets are contained in the HP OpenView Smart Plug-In for Remedy Action Request System integration product bundles.

Table 2-5   **The Products in the OV SPI for Remedy SD Bundles**

SD Product	SD File Set	Description
SPI-REM-SRV	SPI-REM-SERVER	OV SPI for Remedy server software
SPI-REM-CFG	SPI-REM-CONFIG	OV SPI for Remedy Configuration
SPI-REM-DOC	SPI-REM-DOCUM	OV SPI for Remedy Documentation

Table 2-6 describes the contents of the HP OpenView Smart Plug-In for Remedy Action Request System integration filesets.

Table 2-6   **The OV SPI for Remedy SD File Sets**

SD File Set	Description
SPI-REM-SERVER	OV SPI for Remedy server software
SPI-REM-CONFIG	OV SPI for Remedy Configuration including: • OV SPI for Remedy server configuration files • OVO/Unix GUI Integration in uploadable format • OVO/Unix templates and monitor scripts
SPI-REM-DOCUM	OV SPI for Remedy Documentation including • Administrator's Reference Guide • Software Release Notes

Installing the OV SPI for Remedy Software

The HP OpenView Smart Plug-In for Remedy Action Request System integration is available on the HP OpenView Operations for Unix SPI CD-ROM.

There are two CDs in the Applications CD-ROM set: The first contains the products' management server install packages. The second contains the *OpenView Reporter and OpenView Performance Manager* integrations for the applications. For the OV SPI for Remedy, use the first CD.

To install the HP OpenView Smart Plug-In for Remedy Action Request System integration software bundle on the OVO/Unix management server, carry out the following steps:

1. Login to the OVO/Unix management server as **root** user.

2. If not already present, create a directory to mount the CD-ROM:

> `mkdir /<mount_point>`
>
> For example: `mkdir /cdrom`

3. Mount the CD-ROM as user root. On HP-UX, enter:

   `mount -r -F cdfs /dev/<cdrom_drive_name> /<mount_point>`

   For example, for a local CD-ROM on HP-UX you might enter:

   `mount -r -F cdfs /dev/dsk/c0t2d0 /cdrom`

   On HP-UX, you can also run SAM and mount the CD-ROM to a specific path in the `Disks and File Systems` window.

4. The command to install the software depends on the platform of the management server (and the mountpoint of the CD-ROM).

   • For a **HP-UX 11.x** management server, enter:

     `swinstall -s /cdrom/OV_DEPOT/11.0HPUX.sdtape SPI-Remedy`

   • For a **Solaris** management server, enter:

     `swinstall -s /cdrom/OV_DEPOT/SOLARIS.sdtape SPI-Remedy`

   The `swinstall` command installs the `SPI-Remedy` software bundle from the depot and performs basic configuration. The software bundle contains all the OV SPI for Remedy software, configuration files, and documentation.

   On HP-UX, you can use the `swinstall` GUI by omitting the specification of the product or bundle which is to be installed.

NOTE   The products `SPI-REM-SRV` and `SPI-REM-CFG` in the `SPI-Remedy` bundle contain a dependency on the OVO/Unix file set `OVOPC.OVOPC-UX-MGR78` (the OVO/Unix server software) to prevent installation on systems without OVO/Unix installed.

To install *only* the OV SPI for Remedy documentation, change the software bundle parameter to `SPI-RemedyDoc`.

Verifying Installation of the OV SPI for Remedy Software

To verify that the installation of the HP OpenView Smart Plug-In for

---

Remedy Action Request System integration completed successfully on the OVO/Unix management server, carry out the steps in the following list:

1. On the command line, enter: **swlist**

   Look for the following entries:

   SPI-Remedy A.02.50

   SPI-RemedyDoc A.02.50

2. In the OVO/Unix GUI, check that the following elements have been added:

Node Groups	Remedy SPI Servers
	AR Servers NT
	AR Servers UX
Message Groups	Remedy-SPI
	AR System
Application Groups	Remedy SPI Admin
	Remedy SPI
Users	remspi_adm
	remspi_op
Message-Source Templates	OV SPI for Remedy - AR Server NT
	OV SPI for Remedy - AR Server UX
	OV SPI for Remedy - ITO

3. If either of these steps fails to produce the desired results, use the swverify command to ensure that all rules and dependencies were obeyed during the installation of the OV SPI for Remedy software. Enter:

   swverify -x autoselect_dependencies=false SPI-Remedy

   a. Check the following log files for more specific information relating to installation problems:

      - /var/adm/sw/swagent.log

      - /var/adm/sw/swinstall.log

b. Check your installation with the details in Table 2-8, "File Locations on the OVO/Unix Management Server," on page 51.

Assigning the OV SPI for Remedy Templates

In order to facilitate the assignment of the OV SPI for Remedy templates to the various OVO/Unix and AR servers in your environment, you first need to add the servers to the OVO/Unix Node Bank window and then copy the nodes to the appropriate default node groups, which are created during the installation of the OV SPI for Remedy software.

1. Start the OVO/Unix GUI.

2. In the Node Bank window, select the following menu option:

   Actions:Node-> Add

   The Add Node window opens.

3. Enter the requested details as appropriate (label, long hostname, etc), click the [IP Address] button to resolve IP address automatically, and ensure the newly added nodes appear correctly in the Node Bank window.

   Repeat this step for each OVO/Unix and AR server in your environment that you want to monitor with the OV SPI for Remedy.

4. Open the Node Group Bank window

5. Expand the appropriate node group (AR Servers NT, AR Servers UX, or Remedy SPI Servers)

6. Drag the newly added nodes (representing your OVO/Unix and AR servers) from the Node Bank window and drop them into the appropriate Node Group window. For example, drag an NT node from the Node Bank window and drop it into the AR Servers NT node group window.

The installation of the OV SPI for Remedy software automatically assigns the appropriate template group to the corresponding node group as illustrated in Table 2-7. Consequently, any managed node that you copy to the node groups listed has the correct templates assigned

automatically.

Table 2-7   **Automatic OV SPI for Remedy Template-Group Assignment**

Node Group	Template Group
AR Servers NT	OV SPI for Remedy - AR Server NT
AR Servers UX	OV SPI for Remedy - AR Server UX
Remedy SPI Servers	OV SPI for Remedy - ITO

Distributing the OV SPI for Remedy Templates

To distribute the OV SPI for Remedy templates to the OVO/Unix management server and, assuming it is to be a OVO/Unix managed node, the AR server, too:

1. In the Node Group Bank Window, click the AR Servers NT, AR Servers UX, and Remedy SPI Servers node groups and select the following menu option:

   Actions:Agents-> Install/Update S/W & Config...

   The Install/Update S/W & Config... window opens.

2. Verify that the correct nodes appear and that the template assignments you made in the previous step are correctly displayed in the Assigned Templates list.

3. Next, check (✓) the items Templates, Actions, Commands, Monitors, and click [OK].

Installed File Locations

The installation process copies the necessary files to the OVO/Unix management server. Table 2-8 shows the directories that the installation process creates on the OVO/Unix management server for the OV SPI for

Remedy server.

Table 2-8   **File Locations on the OVO/Unix Management Server**

File Type	Directory
Binaries	`/opt/OV/bin/remspi`
Temporary and runtime	`/var/opt/OV/share/tmp/remspi`
Log files	`/var/opt/OV/log/remspi`
Configuration	`/etc/opt/OV/share/conf/remspi`
OVO/Unix integration	`/var/opt/OV/share/tmp/OpC_appl/remspi`
Documentation	`/opt/OV/doc`
Libraries	`/opt/OV/lib/libremspi.sl`
	`/opt/OV/lib/libspi.sl`

The installation process also creates two message-catalog files for the OV SPI for Remedy, namely; remspi.cat and remspii.cat. The remspii.cat file contains the instruction text for the OV SPI for Remedy messages. Both files can be found on the OVO/Unix management server in the following location:

`/opt/OV/lib/nls/C/`

Table 2-9 shows the directories for OV SPI for Remedy clients that you install on a OVO/Unix managed node. These are the standard locations for OVO/Unix commands.

Table 2-9   **File Locations for the Installed OV SPI for Remedy Client**

Operating System	Directory
UNIX	`/var/opt/OV/bin/OpC/cmds`
NT	`\usr\OV\bin\OpC\cmds`

Upgrading the OV SPI for Remedy Software

This section describes how to upgrade the HP OpenView Smart Plug-In for Remedy Action Request System integration software from version A.01.x to the current version, A.02.50. The upgrade requires two steps, each of which is described in detail in the sections that follow:

- Installing the New Software
- Configuring the New Software

Installing the New Software

Version A.02.50 of the OV SPI for Remedy software can be installed on an OVO/Unix server system even if an A.01.* version of the Remedy SPI is already installed. The installation of the *new* version of the software does not modify the *installed* software. The installed version of the OV SPI for Remedy may be removed after you have completed and verified the upgrade.

NOTE   Since the new version of the HP OpenView Smart Plug-In for Remedy Action Request System integration uses the same port as version A.01.* of the OV SPI for Remedy, you will need to stop the A.01.* OV SPI for Remedy server processes before you start the installation of the new OV SPI for Remedy A.02.50 software. However, you can leave the A.01.* installation in place until the new A.02.50 version has been customized and tested.

1. Stop the A.01.* OV SPI for Remedy server processes on the OVO/Unix management server. Enter:

   ovstop ITAROS_itosrv ITAROS_arssrv

2. On the OVO/Unix management server, delete the ovstart(1) registration in order to prevent any accidental or unintended start-up of the A.01.* servers after the installation of the new software. Enter:

   **ovdelobj /opt/OV/OpC/integration/itaros/config/\\
   ITAROS_itosrv.lrf**

   ovdelobj /opt/OV/OpC/integration/itaros/config/

ITAROS_arssrv.lrf

3. Install the new version of the software: follow the instructions in "Installing the OV SPI for Remedy" on page 44.

Configuring the New Software

The HP OpenView Smart Plug-In for Remedy Action Request System integration A.02.50 provides a utility which automatically converts the A.01.* configuration to the new format and location required by version A.02.50 of the OV SPI for Remedy. This utility creates the default OV SPI for Remedy configuration file and rules file from the A.01.*configuration file. Note that you will have to re-enter passwords since there is no decryption utility. To run the upgrade automatic-configuration utility, enter:

/opt/OV/bin/remspi/remspi_upgr_v1.sh

The automatic-configuration utility allows you to re-create automatically all the ARS filters which forward AR modifications to OVO/Unix. Similarly, you can also use the automatic configuration utility to remove ARS filters created for an A.01.* installation if they are no longer needed with the new software installation. Both tasks use the utility program remspifilter. For more information, see ARS Filters and Active Links and "Configuring the ARS Filter Automatically" on page 150.

NOTE   Note that the automatic configuration of the ARS filters is only available for filters not depending on the tag IF_MODIFIED

3 Getting Started

This section describes how to configure the HP OpenView Smart Plug-In for Remedy Action Request System integration. The section covers the following areas:

- Uploading the Example Form to ARS

- Configuring Communication Between OVO/Unix and ARS

- Configuring Communication between ARS and OVO/Unix

Overview

This chapter uses a simple scenario to take you through the steps required to set the communication between OVO/Unix and the ARS via the HP OV SPI for Remedy. The instructions in this section assume that you have already successfully installed the Remedy SPI software on the OVO/Unix management server and assigned and distributed the Remedy SPI templates to both the OVO/Unix management server and the AR server. For more information about the installation of the software and the assignment and distribution of the Remedy SPI Templates, see "Installing the OV SPI for Remedy" on page 44.

The configuration of the Remedy SPI involves two simple steps which establish and configure communication channels between OVO/Unix and ARS via the HP OV SPI for Remedy. Once OVO/Unix and ARS can communicate with each other, you can use OVO/Unix messages to create Action Requests in ARS, and configure ARS to update the original OVO/Unix message in the event of any change to the status of an Action Request created by a OVO/Unix message.

To configure communication between OVO/Unix and ARS, you need to carry out the following high-level steps in the order specified:

1. Configure the OVO/Unix management server to communicate with your ARS server(s) via the HP OV SPI for Remedy server

2. Configure the ARS server(s) to communicate with the OVO/Unix management server via the HP OV SPI for Remedy client

To help you understand how the configuration process works and allow you to verify whether or not the configuration has completed successfully, the Remedy SPI provides a very simple example AR form, which you can upload to the AR server and quickly configure. For ease of use, the same form appears elsewhere in the book, for example; in Chapter 5 , "Customizing the OV SPI for Remedy."

Uploading the Example Form to ARS

In this step, you upload a simple form, provided as part of the Remedy SPI package, from the OVO/Unix management server to the AR server. Once the form is uploaded to ARS, you can use it to help you understand how the communication channels between OVO/Unix and ARS work. In the process, you see how OVO/Unix and ARS use the Remedy SPI to communicate with each other and what has to be done to set up the any other forms you have already created or intend to create.

To upload the example ARS form *Problem Report* to ARS:

1. Depending on the operating system installed on the AR server, use the ftp command to transfer either the ProblemReport-NT-vpo6.def or the ProblemReport-UX.def file from the OVO/Unix management server to the AR server. You can find the files on the management server in the following location:

   /etc/opt/OV/share/conf/remspi/gui/

2. In the ARS administrator's GUI on the AR server, select the AR server into which you want to import new definitions and import the form using the following menu sequence:

   Tools:Import Definitions

3. Select the definitions file to import, in this case:

   ProblemReport-NT-vpo6.def *or* ProblemReport-UX.def

4. In the Import Files window, choose the Select All option and click the OK button.

5. Open the Forms property sheet to verify that the form was successfully imported.

Configuring Communication Between OVO/Unix and ARS

In this step, you set up and configure the connection between OVO/Unix and the Remedy ARS. This involves the following high-level steps, each of which is described in detail in the sections that follow:

- Create a HP OV SPI for Remedy login to OVO/Unix so that OVO/Unix can use the HP OV SPI for Remedy to forward messages to the ARS. This involves:

  1. Adding a new OVO/Unix user (for the HP OV SPI for Remedy) to the `User Bank` window in the OVO/Unix GUI

  2. Set up the same user in the HP OV SPI for Remedy-server configuration scripts

- Configure the HP OV SPI for Remedy so that it knows to which AR servers to send the messages it receives from OVO/Unix, what conditions apply, and what schemas exist in ARS

- Test the connection you have set up between OVO/Unix and ARS

For more information on adding and modifying users in OVO/Unix, see "HP OV SPI for Remedy Users" on page 77. For information on how to set up the Trouble-Ticket (TT) interface in OVO/Unix so that OVO/Unix messages can be automatically submitted to the ARS via the HP OV SPI for Remedy, see "Setting up the Trouble-Ticket Interface" on page 93.

Create a Login to OVO/Unix for the Remedy SPI

To create an ARS login to OVO/Unix, you need to create a new OVO/Unix user with the appropriate permissions in OVO/Unix and copy the details of the new OVO/Unix user, including an encrypted version of the new user's password, into the `remspi.cfg` file, which you can find in the following location on the OVO/Unix management server: `/etc/opt/OV/share/conf/remspi/remspi.cfg`.

If you want to use the `FORCE OWN` or the `DISOWN` keywords in the *rules* file, you will have to configure the *encrypted* password of the OVO/Unix administrator in this file, too.

NOTE

The new OVO/Unix user you create in this step is the user the Remedy SPI server will to use to access the OVO/Unix database in order to fetch the message attributes or to modify messages. Consequently, the user you create must have the necessary permissions to see *all* the messages that you want the HP OV SPI for Remedy to forward from OVO/Unix to ARS. For more information on user privileges in OVO/Unix, see the section on the responsibility matrix in "HP OV SPI for Remedy Users" on page 77 and the *HP OpenView VantagePoint Operations Concepts Guide*.

To create a new OVO/Unix user login for the Remedy SPI server:

1. Use the OVO/Unix GUI to create a OVO/Unix user with access to the HP OV SPI for Remedy applications, node groups and, in addition, the messages in the OVO/Unix database:

   a. First make a copy of an existing HP OV SPI for Remedy user by opening the OVO/Unix User Bank window, right-clicking an existing user such as remspi_adm, and saving the copied user under a new name. For example; rempsi_server.

   b. Modify the new user by right-clicking the newly created user in the OVO/Unix User Bank window and selecting the Modify... option from the pop-up menu.

   c. Use the OVO/Unix Responsibility Matrix window to ensure that the new user has access to the appropriate HP OV SPI for Remedy node groups and the relevant HP OV SPI for Remedy message groups.

2. Type in a password for the new OVO/Unix user you have set up

3. Encrypt the password you have just created using the remspipasswd command. For example, on the command line in a separate shell, enter:

   remspipasswd <*new-user_password*>

4. Open the /etc/opt/OV/share/conf/remspi/remspi.cfg file and enter (by typing or cutting and pasting) the appropriate values for:

   REMPSI_ITO_USER

   The name of the new user you created in OVO/Unix in step 1

REMSPI_ITO_PASSWD

The encrypted password you received in step 3 using the
remspipasswd command

REMSPI_ITO_ADMIN_PASSWD

The encrypted password for the opc_adm OVO/Unix user you
received using the remspipasswd command with the password of
the OVO administrator (note that this line is optional and only
needed if you want to make use of the FORCE OWN/DISOWN
features).

Example 3-1   **Example Configuration File**

```
# File:        remspi.cfg
# Description: Configuration file for Remedy(tm) SPI
# Package:     HP OpenView SMART Plug-In for
#              Remedy(tm) Action Request System(r)
# Note: Value must be on same line as keyword

REMPSI_ITO_USER     remspi_server
REMSPI_ITO_PASSWD   C338D5F21219E076C2000C45AA0475925A1306EA
REMSPI_ITO_ADMIN_PASSWD C338D5F21219F076C2000C4VAA0475925A1306EA
```

5. Save the modifications to the remspi.cfg file

NOTE   Note that the remspi.cfg file which is delivered with the HP OV SPI
for Remedy contains the encoded default password of the opc_adm user of
a OVO/Unix 7.* system. This entry has to be changed for a production
system or a OVO/Unix A.08.00 system.

Configure the HP OV SPI for Remedy in OVO/Unix

In this section, you tell OVO/Unix about the AR servers to which it has to
send messages, what (if any) conditions apply, and what forms exist in
ARS that need to be modified by OVO/Unix via the Remedy SPI. You
define these details in the rules file, which you can find in the following
place on the OVO/Unix management server:
/etc/opt/OV/share/conf/remspi/rules. For more information about
the structure of the rules file and the syntax you are expected to use, see
"Customizing the Rules File" on page 95.

NOTE   If you simply want to understand how the example schema works, you do
not need to change or modify values in the specified in the
MSGCONDITIONS section of the rules file and described in step 3 below.
However, this information is vital when you need to set up your own
conditions at a later date.

To configure the HP OV SPI for Remedy in OVO/Unix:

1. Open the `rules` file with a text editor

2. Enter or modify the following values in the `TARGET` section:

   SERVER   Name of the AR server where *SCHEMA* is defined. This value is *mandatory*: no default value is possible.

   SCHEMA   Name of the form (schema) to be modified by the OVO/Unix message

   USER   User name to log in to the AR server where *SCHEMA* is defined. This is the account the Remedy SPI server uses to log in to the AR server. Note that you should use different account names for the Remedy SPI server and for interactive AR users in order to distinguish Remedy SPI server triggered transactions from normal AR user transactions.

   PASSWORD   Password of *USER*

3. Modify the following values in the `MSGCONDITIONS` section:

   DESCRIPTION   of the condition

   CONDITION   <empty>

   SET TARGET   Name of the form to be linked to the OVO/Unix message generated by the condition match

4. Save the modifications to the `rules` file.

5. On the OVO/Unix management server, restart the HP OV SPI for Remedy server process to activate the new configuration. Enter:

   ovstop RemedySPI

   ovstart RemedySPI

Verifying the Configuration

To verify that your configuration has indeed established the recommended means of communication between OVO/Unix and the ARS via the HP OV SPI for Remedy, you can use the HP OV SPI for Remedy to send a OVO/Unix message to the ARS and check whether or not an Action Request has been created.

1. In the OVO/Unix GUI, select a message in the `Message Browser` window.

2. In the OVO/Unix GUI, click the `Submit Message(s)` application,
   which you can find in the `Remedy SPI` application group in the
   `Application Bank` window

3. Verify that the OVO/Unix message you send creates an Action
   Request in the ARS GUI:

   a. Start the ARS User GUI

   b. Find the example schema using the following menu sequence:

      `File:Search For-> Problem Report System`

   c. Click the `Search` button

   d. Select the message you sent from OVO/Unix from the list at the
      top of the window

   e. Verify the values in the AR schema that is displayed at the bottom
      of the window

Configuring Communication between ARS and OVO/Unix

In this step, you set up and configure the connection between the Remedy AR server and OVO/Unix. When this connection is up and running, ARS can use the HP OV SPI for Remedy to inform OVO/Unix about any changes to Action Requests that have been linked to or perhaps even created by OVO/Unix messages. Setting up the connection between ARS and OVO/Unix involves the following high-level steps:

- Configure the `itoupdate` command in such a way that OVO/Unix message flags such as own/disown, acknowledge, escalate, and so on, can be set automatically as a result of specific changes to the status of AR fields

- Register ARS filters so that they can use the `itoupdate` command to update OVO/Unix messages

Note that the `itoupdate` command is installed on the AR server either when you install the OV SPI for Remedy software on the AR server (the OVO/Unix managed node) or when you copy the `itoupdate` file from the OVO/Unix management server to the AR server manually. For more information, see "Installing the OV SPI for Remedy Software" on page 46.

Configuring the itoupdate Command

This step describes how to use the `rules` file on the OVO/Unix management server to configure the way in which the `itoupdate` command reacts to ARS filters on the AR server.

NOTE   For the purposes of understanding how the example schema works, you do not need to change or modify values in the rules file. However, this information is vital when you need to modify the `itoupdate` command to suit the requirements of your own environment.

To configure the `itoupdate` command:

1. Open the rules file on the OVO/Unix management server:

`/etc/opt/OV/share/conf/remspi/rules`

2. In the `TARGET` section, find the `ITO_UPDATE` sub-section

3. In the `ITO_UPDATE` section, add entries as appropriate for the following values:

   - `OWN/DISOWN`

   - `ACKNOWLEDGE/UNACKNOWLEDGE`

   - `ESCALATE`

     "An Example of a Rules File" on page 136 shows you a working rules file. This is the rules file that is shipped with the HP OV SPI for Remedy software and which you installed in "Uploading the Example Form to ARS" on page 58. For more detailed information about linking changes to Action Request fields with changes to the status of OVO/Unix messages, see "Defining Update Rules" on page 100.

Registering the ARS Filters

This step describes how to register ARS filters on the OVO/Unix management server so that they can use the `itoupdate` command to update OVO/Unix messages with information relating to any change of status of Action Requests. For more information about the `remspifilter` command and the options it accepts, see "The remspifilter Command" on page 148

To register an ARS filter:

1. On the OVO/Unix management server, stop the HP OV SPI for Remedy server process. Enter:

   ovstop RemedySPI

2. Use the command `/opt/OV/bin/rempsi/remspifilter` to register the ARS filters on the OVO/Unix management server. Enter:

   **/opt/OV/bin/rempsi/remspifilter -target <*target_name*>\
   -create**

   Where for the example schema you have uploaded; `target_name` = `ProbSys`. For more information about targets and how to define them, see "Defining Targets" on page 99.

3. Confirm the activation of the HP OV SPI for Remedy filter. Enter:

```
Confirm Creation (y/n) [y]: y
```

4. On the OVO/Unix management server, restart the HP OV SPI for Remedy server process to activate the new configuration. Enter:

   ovstart RemedySPI

NOTE   Note that the automatic generation of AR filters is only available for filters not using the IF_MODIFIED tag. These filters have to be created manually using the AR Admin GUI.

Communicating with the AR Server through a Firewall

If there is a firewall between the AR server and the OVO management server or the communication port of the AR server has been restricted otherwise, the Remedy SPI server can be instructed to use a special, fixed port to communicate with the AR server.

In the file /etc/opt/OV/share/conf/remspi/remspi.cfg, insert a line "REMSPI_ARTCPPORT *portnum*", where *portnum* is the number of the port the AR server listens for requests.

NOTE   Note that this situation requires configuration on the AR server, too. For details, see the "Remedy Action Request System Administrator's Reference, Chapter 3".

Using private queues on the AR Server

If a private queue should be used for the SPI server communicating with the AR server, the number of this rpc queue can be specified.

In the file /etc/opt/OV/share/conf/remspi/remspi.cfg, insert a line "REMSPI_ARRPC *rpcnum*", where *rpcnum* is the number of the rpc port used by the AR server.

NOTE   Note that this situation requires configuration on the AR server, too. For details, see the "Remedy Action Request System Administrator's

Reference, Chapter 3".

Verifying the Configuration

To verify that you have successfully configured the `itoupdate` command to update OVO/Unix messages automatically as a result of changes to the status of Action Requests, you can use the ARS to modify the Action Request you created when you send a OVO/Unix message in "Verifying the Configuration" on page 63. To verify that the addition or update of comments to an Action Request are reflected in the annotations attached to the original OVO/Unix message:

1. Submit a OVO/Unix message to ARS as described in "Verifying the Configuration" on page 63

2. Login to the AR server and open the `Problem Report` Schema

3. In ARS, select the Action Request created by the OVO/Unix message you have just submitted and open the Action Request for modification

4. Enter some text in the `Work Log` field and save the changes

5. In OVO/Unix, open the `Message Browser` window, select the message you originally submitted to ARS, and click the `[Annotations]` button.

   The `Message Annotations` window opens allowing you to check for the presence in the list of annotations of the text you entered in the `Work Log` field in step 4 above.

To verify that changes to the status of Action Request in ARS are automatically reflected in a corresponding change of status to the appropriate messages in OVO/Unix:

1. In ARS, select the Action Request created by the OVO/Unix message you have just submitted and open the Action Request for modification

2. Change the status of the Action Request to `Assigned`

3. In OVO/Unix, open the `Message Browser` window, select the message you originally submitted to ARS and ensure that the message is now owned by the appropriate HP OV SPI for Remedy user.

4. In ARS, change the status of the same Action Request to `Closed`

5. In OVO/Unix, open the `History Message Browser` window, select

the message you originally submitted to ARS and ensure that the
message is now acknowledged by the appropriate HP OV SPI for
Remedy user.

4 Using the HP OV SPI for Remedy

This section describes the HP OV SPI for Remedy interface that you will use in the performance of your every-day tasks and gives advice and information concerning how to go about setting up and modifying the main aspects. The section covers the following areas:

- HP OV SPI for Remedy Node Groups
- HP OV SPI for Remedy Message Groups
- HP OV SPI for Remedy Users
- HP OV SPI for Remedy Application Groups
- HP OV SPI for Remedy Administrator Tools
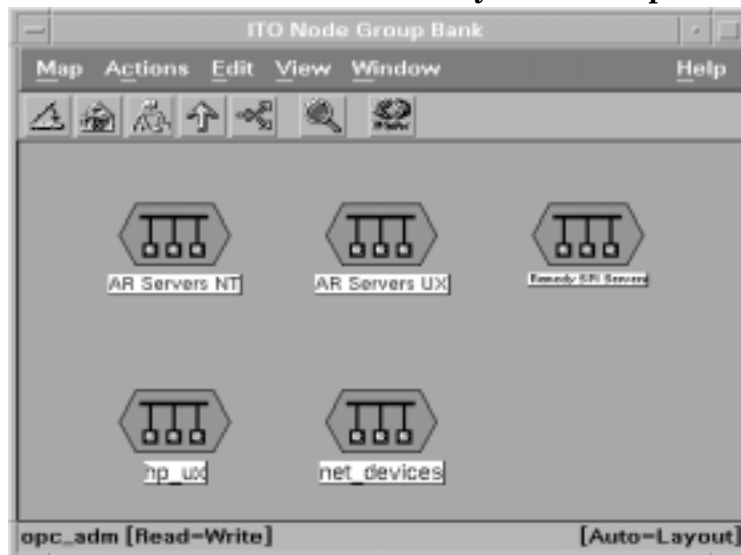- HP OV SPI for Remedy Operator Tools

HP OV SPI for Remedy Node Groups

Figure 4-1 shows the node groups that are installed by default with the HP OpenView Smart Plug-In for Remedy Action Request System integration. The node groups are:

- AR Servers
- Remedy SPI Servers

Figure 4-1   **The Default HP OV SPI for Remedy Node Groups**

The AR servers in your ARS environment should be added to the AR Servers NT and AR Servers UX node group as appropriate either by dragging the managed node from the Node Bank window and dropping it onto the appropriate icon in the Node Group Bank window or by opening the appropriate node group and using the following menu sequence:

Actions:Node-> Add...

The systems running the HP OV SPI for Remedy server, normally the OVO/Unix management servers, should be added to the node group Remedy SPI Servers in the same way.

The installation of the OV SPI for Remedy software automatically

---

Chapter 4   **73**

assigns the appropriate template group to the corresponding node group as illustrated in Table 2-7 on page 50. Consequently, the managed nodes that you copy to the node groups listed has the correct templates assigned automatically.

For more information about which platforms, operating systems, and AR versions that are supported by the HP OpenView Smart Plug-In for Remedy Action Request System integration, see "Installation Prerequisites" on page 41.

NOTE   Node groups are assigned to message groups: message groups are assigned to users.

The node groups; `AR Servers NT`, `AR Servers UX,` and `Remedy SPI Servers` are assigned by default to the message groups; `AR System` and `Remedy-SPI`. You can then assign these message groups to the HP OV SPI for Remedy users you add and make responsible for your AR systems and the HP OV SPI for Remedy in general. In this way, the messages generated by HP OV SPI for Remedy systems are sent automatically to the HP OV SPI for Remedy users whom you make directly responsible for these systems. For more information about HP OV SPI for Remedy users, see "HP OV SPI for Remedy Users" on page 77. For more information about HP OV SPI for Remedy message groups, see "HP OV SPI for Remedy Message Groups" on page 75.

HP OV SPI for Remedy Message Groups

This section describes the message groups that are installed by default with the HP OpenView Smart Plug-In for Remedy Action Request System integration. The message groups are:

- AR System

- Remedy-SPI

You assign the message groups; AR System and Remedy-SPI to the HP OV SPI for Remedy users whom you add and make responsible for your AR systems and the HP OV SPI for Remedy in general.

Figure 4-2　　　　**The Default HP OV SPI for Remedy Message Groups**

Since the message groups; AR System and Remedy-SPI are assigned by default to the node groups; AR Servers NT, AR Servers UX, and Remedy SPI Servers, any messages originating from the HP OV SPI for Remedy systems present in the node groups; AR Servers NT, AR Servers UX, and Remedy SPI Servers are sent automatically to the HP OV SPI for Remedy users who are directly responsible for the nodes in these node groups. For more information about HP OV SPI for

Remedy users, see "HP OV SPI for Remedy Users" on page 77. For more information about HP OV SPI for Remedy node groups, see "HP OV SPI for Remedy Node Groups" on page 73.

HP OV SPI for Remedy Users

This section describes the default HP OV SPI for Remedy users that appear in the User Bank window after you have installed the HP OpenView Smart Plug-In for Remedy Action Request System integration. The users are:

- remspi_adm
- remspi_op

Figure 4-3   **The HP OV SPI for Remedy User Bank**

The default OVO/Unix user accounts for the HP OV SPI for Remedy, rempsi_adm and rempsi_op are intended as examples: they can be used as a basis for the creation of "real" OVO/Unix user accounts, which are added by the OVO/Unix administrator. The default HP OV SPI for Remedy user accounts have access to the default HP OV SPI for Remedy applications in order to allow them to manage the HP OV SPI for Remedy (starting and stopping AR servers, for example) and to work with the HP OV SPI for Remedy (submitting and attaching messages). In addition, the default HP OV SPI for Remedy users can "see" the default HP OV SPI for Remedy node groups so that they can select the managed nodes on which the application will run.

To use the default users as a basis for creating your own users, select the

user you want to copy, right-click the mouse button and select `Copy...` from the menu that pops up. Once you have copied the user, you can select the newly created user and use the same procedure to modify the new user's settings. Click the [`Responsibilities...`] button to display the `Responsibility Matrix` window illustrated in Figure 4-4 on page 78.

The OVO/Unix account you set up for the HP OV SPI for Remedy server in "Configuring Communication Between OVO/Unix and ARS" on page 59 and which is defined in the configuration file `remspi.cfg` is used by the HP OV SPI for Remedy server to access messages in the OVO/Unix database. Clearly, this account must be granted access to all the messages which should be forwarded to the AR system.

Figure 4-4   **Modifying the HP OV SPI for Remedy User's Responsibilities**
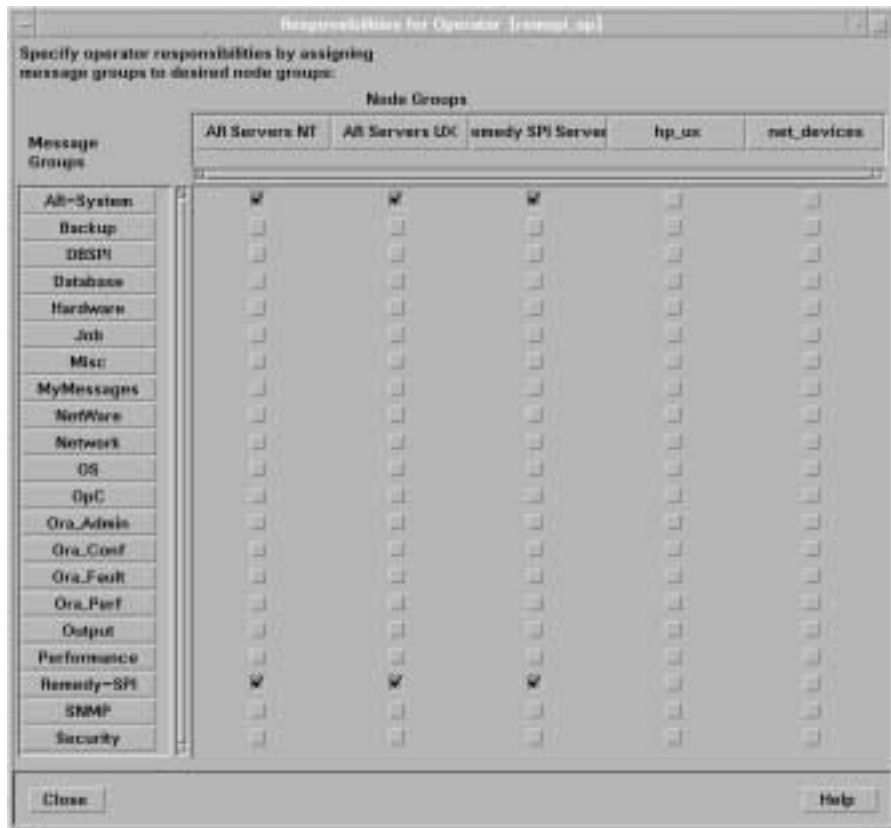
You need to pay particular attention to which node groups are assigned

to which message groups. In the example illustrated in Figure 4-4, "Modifying the HP OV SPI for Remedy User's Responsibilities," the HP OV SPI for Remedy user, `remspi_op`, will see all the messages from the message groups `AR System` and `Remedy-SPI` and generated by systems present in the `AR Servers NT,` `AR Servers UX,` and `Remedy SPI Servers` node groups. For more information about node and message groups in the HP OV SPI for Remedy, see "HP OV SPI for Remedy Node Groups" on page 73 and "HP OV SPI for Remedy Message Groups" on page 75.

HP OV SPI for Remedy Application Groups

To assist the HP OV SPI for Remedy user in the performance of
every-day tasks, the installation process of the HP OpenView Smart
Plug-In for Remedy Action Request System integration creates two new
application groups specifically for the HP OV SPI for Remedy in the
OVO/Unix application bank. The new application groups are:

`Remedy SPI`	contains applications for operating the HP OV SPI for Remedy
`Remedy SPI Admin`	contains applications for the administration of the HP OV SPI for Remedy

Figure 4-5 shows the two HP OV SPI for Remedy-specific application
groups in the `OVO/Unix Application Bank` window.

Figure 4-5   **Default HP OV SPI for Remedy Application Groups**

NOTE   Your organization may restrict access to the application groups, or create different ones.

For more information about the contents of the Remedy SPI and Remedy SPI Admin application groups, see "HP OV SPI for Remedy Operator Tools" on page 86 and "HP OV SPI for Remedy Administrator Tools" on page 82, respectively.

HP OV SPI for Remedy Administrator Tools

The application group Remedy SPI Admin contains applications that
enable the HP OV SPI for Remedy administrator to manage the HP OV
SPI for Remedy from within OVO/Unix. Figure 4-6 shows the Remedy
SPI Admin application group, which by default contains the following
applications:

- SPI Server Status

- Stop SPI Server

- Start SPI Server

- SPI Test

- Stop AR Server UX

- Stop AR Server NT

- Start AR Server UX

- Start AR Server NT

**Figure 4-6**   **Administrator Applications**
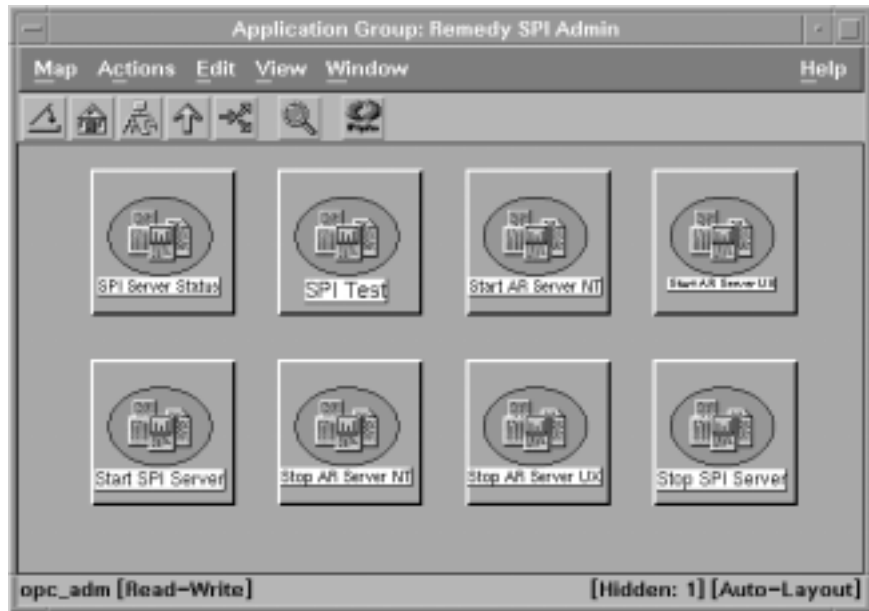
NOTE   Remedy SPI Admin is the default application group that the installation
procedure creates for a HP OV SPI for Remedy administrator. You may
add, modify, or group the applications according to the requirements of
your organization.

The HP OV SPI for Remedy Server's Status

You can check the status of the HP OV SPI for Remedy server by
double-clicking the SPI Server Status application. The status
information appears in an Output of Application window.

Figure 4-7 shows example status information that appears when a HP
OV SPI for Remedy server is running normally.

Figure 4-7   **HP OV SPI for Remedy Server Status**
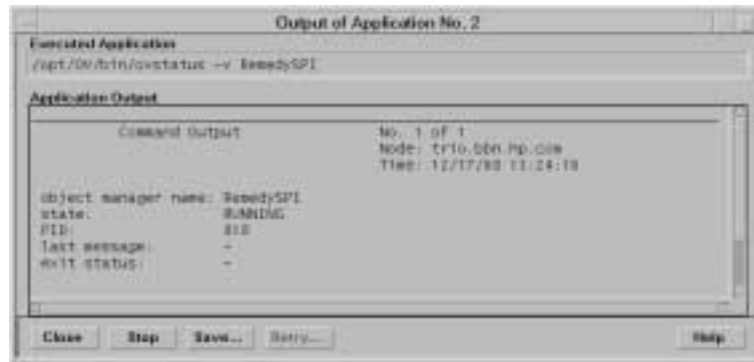
Starting and Stopping the HP OV SPI for Remedy Server

You can start or stop the HP OV SPI for Remedy server by double clicking the `Start SPI Server` or `Stop SPI Server` applications. The application confirms the new status of the HP OV SPI for Remedy server in an `Output of Application` window.

NOTE   You need to stop and restart the HP OV SPI for Remedy server if you change rules file. For more details, see "Customizing the Rules File" on page 95.

Figure 4-8 shows example output that appears when a HP OV SPI for Remedy server starts successfully.

Figure 4-8   **HP OV SPI for Remedy Server Start-up**
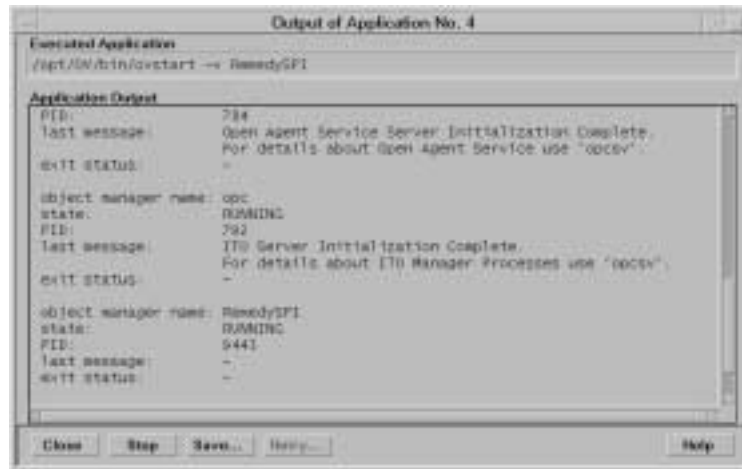
Testing the HP OV SPI for Remedy

As an administrator you can test the status of the Remedy SPI using the
SPI Test application. A message appears in the OVO/Unix Message
Browser window indicating the result of the test.

Starting and Stopping the AR Server

For maintenance purposes, you can start or stop an AR server that runs
on a UNIX managed node. To do this:

1. Select the managed node in the OVO/Unix Node Bank or
   appropriate node group window. (The default node group is AR
   Servers, but this may be different in your organization.)

2. Double click Start AR Server or Stop AR Server application.

HP OV SPI for Remedy Operator Tools

The application group Remedy SPI contains applications that enable the HP OV SPI for Remedy operate to carry out actions from within OVO/Unix. Figure 4-9 on page 86 shows this application group, which by default contains the following applications:

- Submit Message(s)

- Attach Message(s)

NOTE   Remedy SPI is the default application group that the installation procedure creates for a HP OV SPI for Remedy operator. The administrator may add, modify, or group the applications differently according to requirements.

Figure 4-9   **Operator Applications**
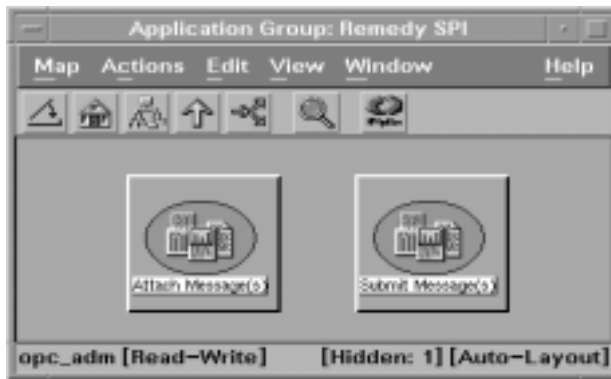
Manually Submitting Messages to an AR System

You can use OVO/Unix messages in the Message Browser window to create action requests in the ARS. The situations in which you submit messages manually depend on how your organization uses OVO/Unix, ARS, and the HP OV SPI for Remedy.

To manually submit OVO/Unix messages to an AR System:

1. In the `Message Browser` window, select one or more messages that you want to submit.

2. In the `Application Bank` window, double-click the `Remedy SPI` application group, and execute the `Submit Message(s)` application.

The application confirms success or failure of your action in an `Output of Application` window. Message submission to ARS fails if:

• The OVO/Unix message does not match any of the conditions in the rules file. For more details, see "Defining Conditions" on page 115.

• OVO/Unix or an OVO/Unix user has already submitted the same OVO/Unix message. You cannot resubmit OVO/Unix messages. The HP OV SPI for Remedy updates them automatically according to the update rules in the rules file. For more details, see "Defining Update Rules" on page 100.

NOTE   If you submit several messages at the same time, the HP OV SPI for Remedy evaluates conditions for the message with the highest severity first. If more than one message has the same severity, the HP OV SPI for Remedy takes the first message it finds with this severity.

If message submission is successful, a new action request appears in the appropriate AR schema. Figure 4-10 shows an example of the output that appears when a user submits an OVO/Unix message successfully.

Figure 4-10   **Manual Message Submission**
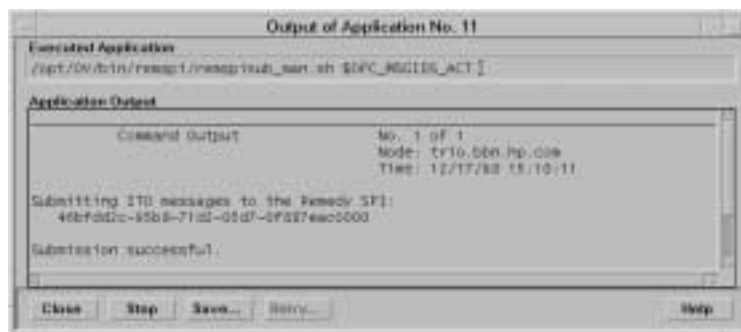
Attaching Messages to Existing Action Requests

You can attach one or more new OVO/Unix messages to an existing action request. You might want to attach several messages to a single action request if one problem causes (or has already caused) several other problems in different, but related, areas.

To attach OVO/Unix messages to an existing action request:

1. In the `Message Browser` window, select the original message for which an action request already exists.

2. Select one or more messages that you want to attach to the action request.

3. Double click the `Attach Message(s)` application.

NOTE   The HP OV SPI for Remedy treats messages attached to an action request as an update to the action request. For more details, see "Defining Update Rules" on page 100.

NOTE   If you want to configure the HP OV SPI for Remedy to be able to perform attachments to existing action request automatically, see "Configuring automatic attachments" on page 134

5 Customizing the OV SPI for Remedy

This section describes how to customize the HP OpenView Smart Plug-in for Remedy to suit the demands of your environment and in order to take full advantage of the functionality of the OV SPI for Remedy in your environment. The section covers the following high-level areas:

- Customizing OVO/Unix
- Customizing the Rules File
- ARS Filters and Active Links
- Customizing the OV SPI for Remedy for Multiple Servers
- Using the OV SPI for Remedy on cluster nodes
- Using External Actions and Data

Customizing OVO/Unix

The OV SPI for Remedy's installation process creates a default configuration within OVO/Unix. You may need to change these, depending on your organization's needs. You can:

- customize objects in the OVO/Unix GUI

- modify the OVO/Unix templates that monitor the availability of the OV SPI for Remedy and AR servers

- set up the Trouble-Ticket interface for the automatic submission of OVO/Unix messages to the ARS via the OV SPI for Remedy

Customizing OVO/Unix GUI Objects

After you install the OV SPI for Remedy, new objects appear in the OVO/Unix GUI that enable you to work with the OV SPI for Remedy. Table 5-1 shows a list of these objects.

Table 5-1   **OV SPI for Remedy Objects in the OVO/Unix GUI**

Object Type	Name	Description
Node Group	OV SPI for Remedy Servers	Nodes on which a OV SPI for Remedy server runs. By default, this contains the current OV SPI for Remedy server node.
	AR Servers NT	Windows NT nodes on which an AR server runs. By default, this window is empty to allow you to add nodes as appropriate.
	AR Servers UX	UNIX nodes on which an AR server runs. By default, this window is empty to allow you to add nodes as appropriate.

Table 5-1   **OV SPI for Remedy Objects in the OVO/Unix GUI**

Object Type	Name	Description
Message Group	Remedy-SPI	OVO/Unix messages that result from monitoring the OV SPI for Remedy server
	AR-System	OVO/Unix messages that result from monitoring an AR server.
Application Group	Remedy SPI Admin	Applications that the OV SPI for Remedy administrator uses to manage the OV SPI for Remedy and AR servers.
	Remedy SPI	Applications that OV SPI for Remedy operators and administrators use to work with the OV SPI for Remedy.
User	remspi_adm	The OV SPI for Remedy administrator has access to all OV SPI for Remedy node, message, and application groups. After installation, the default password is RemspI_adm.
	remspi_op	The OV SPI for Remedy operator has access to all OV SPI for Remedy node and message groups, but only to the OV SPI for Remedy application group. After installation, the default password is RemspI_op.

Modifying the OV SPI for Remedy Templates

You can modify the templates supplied with the OV SPI for Remedy using the Message Source Templates window in the OVO/Unix GUI. For example, you might want to modify the configuration of a template in order to change the polling interval or set up the Trouble-Ticket interface for individual message conditions.

NOTE   Before modifying templates, either back up the default templates or make a copy of the template you want to modify and make any changes to the new copy.

To change or modify a OVO/Unix template:

1. Open the `Message Source Templates` window shown in Figure 5-1 on page 93

2. Expand the template group containing the templates you want to modify

3. Select the appropriate template, and click the `Modify...` button

4. Save the changes you have made and re-distribute the templates to the appropriate nodes

Figure 5-1   **The Message Source Templates Window**
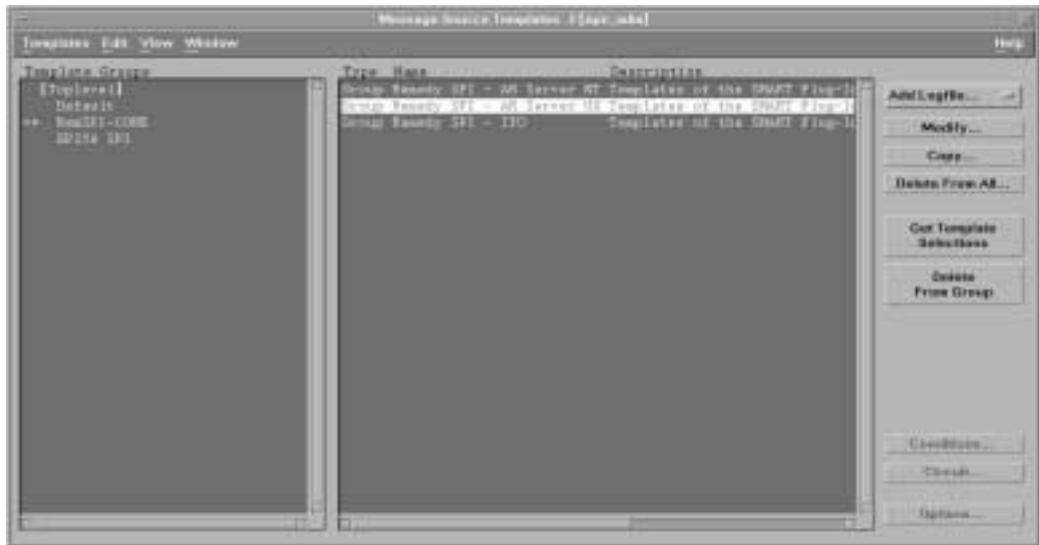
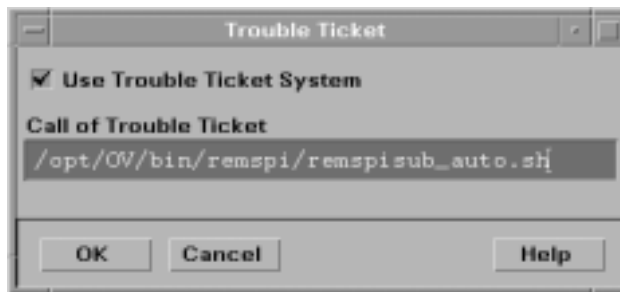
Setting up the Trouble-Ticket Interface

This section describes how to set up the Trouble-Ticket (TT) interface in OVO/Unix so that messages generated by OVO/Unix can be submitted automatically to the AR server via HP OV SPI for Remedy. In addition,

you can also choose to activate the Trouble-Ticket interface for individual message conditions - so that OVO/Unix messages generated as a result of matching specific template conditions are copied to the TT interface.

1. Set up the Trouble-Ticket interface on the OVO/Unix management server:

   a. In the VPO Node Bank window, select the following menu option:

      Actions:Utilities-> Trouble Ticket...

   b. In the Trouble Ticket window that appears:

      - Check the option: Use Trouble Ticket Interface

      - Enter the path to the HP OV SPI for Remedy automatic-submission script:

        /opt/OV/bin/remspi/remspisub_auto.sh

   c. When you are satisfied that the information you have entered is correct, click OK.

Figure 5-2   The Trouble-Ticket Window in OVO/Unix

2. Activate the Trouble-Ticket interface for the individual OVO/Unix message conditions that will generate messages for ARS:

   a. In the OVO/Unix Message Source Templates window, select the template, whose conditions you want to modify.

   b. Click the Conditions... button

   c. In the Actions section of the Condition No window, check the Forward to TT option, and click OK.

   d. Redistribute the templates the AR servers.

Customizing the Rules File

This section describes how to customize the rules file, which determines how the OV SPI for Remedy handles OVO/Unix messages and Action Requests. This section covers the following high-level areas:

- The Structure of the Rules File
- Defining Global Options
- Defining Targets
- Defining Update Rules
- Defining Conditions
- Defining Actions
- An Example of a Rules File

NOTE   After making changes to the rules file, you need to apply the changes by saving the file and stopping and restarting the OV SPI for Remedy server process, which runs on the OVO/Unix management server.

The Structure of the Rules File

The rules file is a plain text file, which resides in the directory /etc/opt/OV/share/conf/remspi. By default, the rules file contains some example rules, which you must edit to meet the needs and requirements of your organization. The rules file defines:

- global options which rule the overall behavior of the OV SPI for Remedy
- targets that describe AR servers and schemas
- update rules that determine how the OV SPI for Remedy updates an OVO/Unix message if the corresponding action request changes
- submit rules that determine how the OV SPI for Remedy updates an OVO/Unix message if the message is submitted as an AR ticket
- conditions that filter the OVO/Unix messages submitted to the OV

SPI for Remedy

- actions that determine how to create or update action requests when a OVO/Unix message satisfies a defined condition

Figure 5-3   **The Structure of the Rules File**

```
File Description

  Global Options
  Target 1
    Update Rules

  Target 2
    Update Rules

  Target n
    Update Rules


  Condition 1
      Action

  Condition 2
      Action

  Condition n
      Action
```

The main body of the rules file, as illustrated in Figure 5-3, starts with an optional global options sections, followed by one or more target definitions, each with its own update rules. You follow these with one or more condition definitions, each with its own action.

The file description consists of:

- the syntax version identifier: SYNTAX VERSION 2.3

- the rules file name that you define as: SPI_RULES "*string*"

- a description that you define as: DESCRIPTION "*string*"

A rules file definition should look similar to the following:

```
SYNTAX VERSION 2.3
 SPI_RULES    "rules"
 DESCRIPTION "Rules file for the AR Problem System"
```

You should note the following important points about the rules file syntax:

- You can insert a comment in the rules file, by preceding it with the hash (#) character. The OV SPI for Remedy server ignores everything between the # character and the end of the line.

- Text strings can contain standard characters including carriage returns and tabs. You can insert print characters within a text string by preceding the print character with two back-slash characters (\\). The same rule applies for if you want to include quotation marks or tab characters. For example:

\\n	represents a new line
\\"	represents a quotation mark
\\t	represents a tab character

- You can also use special variables within the rules file. To do this, always enclose the variable with dollar ($) signs. For example MSG_TEXT. For more information, see Table 5-14 on page 137.

Defining Global Options

A *global option* controls the overall behavior of the HP OV SPI for Remedy.

Table 5-2   **Global Options**

Option	Description
NO_AR_UPDATE	The HP OV SPI for Remedy does not register for changes of OVO messages. Thus, changing an OVO message does not result in a change of the associated AR ticket.

Table 5-2   **Global Options**

Option	Description
ANNO_FAILED_MSG_UPD	If an OVO message could not be updated as a consequence of a change of the associated AR ticket, an annotation is added to this OVO message.
	Note that this added annotation does not trigger an update of the associated AR ticket.
	This option is valid for message updates specified in the ITO_UPDATE and ON_SUBMIT sections of the rules file
BUFFER_TT	This option controls whether the ticket creation and update requests should be buffered if the AR server is down or not available.

Table 5-3 shows the option names that you can define globally.

The following example defines all global options.

```
OPTION NO_AR_UPDATE
OPTION ANNO_FAILED_MSG_UPD
OPTION BUFFER_TT
```

NOTE   If buffering is turned on (OPTION BUFFER_TT), then the requests for creating and updating AR tickets are buffered until the AR server is available again. If the AR server is not available during startup of the SPI server, the SPI server aborts and is not started.

If the AR server is back, the buffered requests are processed in the order they were buffered.

Internal HP OV SPI for Remedy messages are generated when buffering starts and ends.

NOTE	Change events for buffered OVO messages are not buffered.

If more detailed messages regarding the buffering status are needed, the variable REMSPI_QUEUE_CHECK in file remspi.cfg may be used. The value of this variable is treated as a time delay after which the number of buffered requests is checked. If the number of buffered requests is greater than zero, an internal warning message is written to the SPI's log file - and thus to the OVO/Unix message browser. Note that this check is performed regularly, i.e. a message is generated on each check cycle - if buffered requests exists. The number of requests existing in the buffer queues is added to the message.

If the variable is not specified or set to 0, no check is performed.

Defining Targets

A *target* defines the name, location of a form (schema) on an AR server and any login details that are required for the OV SPI for Remedy. This information enables the OV SPI for Remedy server to connect to the AR server and create or change action requests that belong to a particular schema.

Table 5-3　　　　**Target Attributes**

Attribute	Default	Description
TARGET	-	An alias that you use to reference the target schema within the rules file. See the example below.
SERVER	-	The name of the machine where the AR server runs. For example, *ars1.yourcompany.com*. This value is *mandatory*
SCHEMA	-	The name of the form (schema). *This attribute is mandatory.*

Table 5-3   **Target Attributes**

Attribute	Default	Description
USER	-	The user name with which the OV SPI for Remedy server logs in to the AR server. *This attribute is mandatory.*
		Note that you should use different account names for the Remedy SPI server logging in to the AR server and for interactive users in order to be able to distinguish between transactions triggered by the Remedy SPI and transactions triggered by interactive users.
PASSWORD	Blank	The encrypted password for USER.
		Note that this password must be encrypted with the command remspipasswd.

Table 5-3 shows the attributes that you can define for a target. Each attribute must appear once per target definition. You use these attributes to define a target for each of the forms that the OV SPI for Remedy must use.

The following example defines a target for the form Problem Report illustrated in Figure 1-3 on page 23, on a server called ars1.bbn.hp.com.

```
TARGET "ProbSys"
  SERVER    "ars1.bbn.hp.com"
  SCHEMA    "Problem Report"
  USER      "spi_user"
  PASSWORD "C338D5F22121EO76C200F6E6B4E94B2C679B2DA"
```

NOTE   To display the encrypted version of a password, use the command:
/opt/OV/bin/remspi/remspipasswd *<password>*.

Defining Update Rules

Each target has a set of update rules. The update rules define how the OV SPI for Remedy server updates the associated OVO/Unix messages

when it receives details of modifications to an action request.

You define the update rules by adding the ITO_UPDATE attribute to the target definition. You then assign keywords with parameters to the ITO_UPDATE attribute. Table 5-4 on page 102 shows the keywords you use to define for a target. All of the keywords are described in the sections that follow:

- Acknowledging Messages

- Unacknowledging Messages

- Adding Annotations to a Message

- Escalating a Message

- Starting an Operator-Initiated Action

- Taking Ownership of a Message (disowning first on request)

- Removing Ownership from a Message

NOTE   Optional keywords and parameters are listed in brackets, e.g. [IF_MODIFIED *field ID*]

NOTE   The special manually constructed AR filter which is mentioned in the following table is described in section Configuring a special

IF_MODIFIED ARS Filter Manually on page 152.

Table 5-4   **Update Keywords**

Keyword	Parameters	Description
ACKNOWLEDGE	[IF_MODIFIED] *field ID* *value*	Use this keyword to acknowledge the OVO/Unix message if the *field ID* has the given *value*. This also removes the association between the action request and the OVO/Unix message from the OV SPI for Remedy database. If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.
UNACKNOWLEDGE	[IF_MODIFIED] *field ID* *value*	Use this keyword to unacknowledge the OVO/Unix message if the *field ID* has the given *value*. This action is valid only if the message was *locally* acknowledged using the OVO/Unix GUI. If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.

Table 5-4   **Update Keywords**

Keyword	Parameters	Description
ANNOTATE	[IF_MODIFIED *field ID*] *format*	Use this keyword to add an annotation to the OVO/Unix message. The *format* describes the annotation to the OVO message and can consist of text and variables.
		If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.
ANNOTATE	[IF_MODIFIED] *<field ID value>* *format*	Use this keyword to add an annotation to the OVO/Unix message if the *field ID* has a special *value*. The *format* describes the annotation to the OVO message and can consist of text and variables.
		If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.
ESCALATE	*command name*	Use this keyword to escalate the OVO/Unix message if the OV SPI for Remedy client passes the given *command name*.
OP_ACTION	*command name*	Use this keyword to start an operator-initiated action if the OV SPI for Remedy client passes the given *command name*.

Table 5-4   **Update Keywords**

Keyword	Parameters	Description
OWN	[IF_MODIFIED] *field ID* *value*	Use this keyword to take ownership of the OVO/Unix message if the *field ID* has the given *value*. If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.
FORCE OWN	[IF_MODIFIED] *field ID* *value*	Use this keyword to remove the ownership of the message and then to take ownership of the OVO/Unix message if the *field ID* has the given *value* (regardless which user currently owns the message). Note that the REMSPI_ITO_ADMIN_PASSWD keyword is necessary in the remspi.cfg file for this to work. If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.

Table 5-4   **Update Keywords**

Keyword	Parameters	Description
DISOWN	[IF_MODIFIED] *field ID* *value*	Use this keyword to remove ownership of the OVO/Unix message if the *field ID* has the given *value* (regardless which user currently owns the message). Note that in order to use this keyword, the password for the OVO/Unix administrator has to be configured with the REMSPI_ITO_ADMIN_PASSWD keyword in the remspi.cfg file. If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the *field ID* was modified in an AR user transaction and the AR filter was constructed in a special way.

NOTE   If the keyword IF_MODIFIED is used, the AR filter which notifies the Remedy SPI server must be constructed in a special way. This must be done manually using the AR Admin GUI, since the program remspifilter is not able to generate those filters. To learn about these special AR filters, see Configuring a special IF_MODIFIED ARS Filter Manually on page 152..

Acknowledging Messages

The following example shows an update rule that acknowledges the OVO/Unix message if the modified action request has the value Closed in field 7.

```
ITO_UPDATE
  ACKNOWLEDGE 7 "Closed"
```

The next example shows an update rule that acknowledges the OVO/Unix message if the modified action request has the value Closed

in field 7 and this field was modified in the current AR transaction .

```
ITO_UPDATE
  ACKNOWLEDGE IF_MODIFIED 7 "Closed"
```

NOTE
Note that if the IF_MODIFIED tag is used in the update rules, the AR filters which trigger the notification of the Remedy SPI server must be constructed manually in a special way. The program remspifilter is not able to generate these filters on the AR server.   To learn about these special AR filters, see Configuring a special IF_MODIFIED ARS Filter Manually on page 152..

For information on the problems that can occur when acknowledging a large number of messages at the same time, see "Troubleshooting HP OV SPI for Remedy Usage" on page 178

Unacknowledging Messages

The following example shows an update rule that unacknowledges the OVO/Unix message if the modified action request has the value Open in field 7.

```
ITO_UPDATE
  UNACKNOWLEDGE 7 "Open"
```

Adding Annotations to a Message

The following example adds the ITO_UPDATE attribute to the existing target definition and defines an update rule using the ANNOTATE keyword. The ANNOTATE keyword uses text, variables, and control characters. The variables 5 and 1000000001 refer to field identifiers for values in the action request. The control character \\n starts a new line in the annotation.

```
TARGET "ProbSys"
  SERVER "ars1.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "C338D5F22121EO76C200F6E6B4E94B2C679B2DA"
  ITO_UPDATE
    ANNOTATE "$5$ has modified the action request.\\n He or she
made the following work log entry: \\n\\n$1000000001$"
```

The following example shows the different annotations which the OV

SPI for Remedy adds depending on whether field 7 has the value Rejected or Closed.

```
ITO_UPDATE
  ANNOTATE 7 "Rejected" "$5$ rejected the action request."
  ANNOTATE 7 "Closed"   "$5$ closed the action request."
```

The next example shows the same annotations, but now these annotations are added to the OVO message only if the field 7 was modified in the current AR transaction by an interactive AR user. For this example to execute properly, the AR filters which trigger the notification of the Remedy SPI server and thus the evaluation of the update rules, must be constructed manually in a special way. The program remspifilter is not able to create these filters.

```
ITO_UPDATE
  ANNOTATE IF_MODIFIED 7 "Rejected" "$5$ rejected the action
request."
  ANNOTATE IF_MODIFIED 7 "Closed"   "$5$ closed the action
request."
```

You can also use annotation variables in the format that follows an ANNOTATE keyword. Table 5-5 shows the annotation variables that you can use.

Table 5-5   **Annotation Variables**

Variable	Description
ARS_ID	The unique identifier of the modified action request.
ARS_SCHEMA	The schema name that the modified action request belongs to.
ARS_SERVER	The host name of the server where the schema exists.
ARS_USER	The name of the user in the AR System who modified the action request.

The following example shows the ANNOTATE keyword with a format that includes annotation variables.

```
ITO_UPDATE
  ANNOTATE "An action request exists for this message in the
schema $ARS_SCHEMA$ on the server $ARS_SERVER$. The action
request has the ID $ARS_ID$.
```

Escalating a Message

The next example shows an update rule that escalates a OVO/Unix message if an appropriate OVO/Unix escalation manager exists. The command name (here escalate) must match the command parameter in the active link that calls the OV SPI for Remedy client. For more details about setting up an active link, see "Configuring an Active Link" on page 154.

```
ITO_UPDATE
  ESCALATE "escalate"
```

Starting an Operator-Initiated Action

The next example shows an update rule that starts the OVO/Unix message's operator-initiated action, if one exists. The command name (here start-action) must match the command parameter in the active link that calls the OV SPI for Remedy client. For more details about setting up an active link, see"Configuring an Active Link" on page 154.

```
ITO_UPDATE
  OP_ACTION "start-action"
```

Taking Ownership of a Message

The following example shows two update rules that use the OWN keyword. The first rule instructs the OV SPI for Remedy to take ownership of the corresponding OVO/Unix message if the modified action request has the value Open in field 7. Note that taking ownership here is possible only if the message is not owned by another user yet.

The second rule instructs the OV SPI for Remedy to take ownership of the OVO/Unix message (regardless which user owns the message currently) if the modified action request has the value Solved in field 7.

```
ITO_UPDATE
        OWN 7 "Open"
  FORCE OWN 7 "Solved"
```

NOTE   The OVO/Unix user that you specify in the configuration file remspi.cfg becomes the owner of the OVO/Unix messages.

For information on the problems that can occur when taking ownership of a large number of messages at the same time, see "Troubleshooting

HP OV SPI for Remedy Usage" on page 178

Removing Ownership from a Message

The following example shows an update rules that use the DISOWN keyword. The rule instructs the OV SPI for Remedy to remove ownership from the corresponding OVO/Unix message if the modified action request has the value New in field 7.

```
ITO_UPDATE
  DISOWN 7 "New"
```

Defining Submit Rules

Each target has a set of submit rules. The submit rules define how the OV SPI for Remedy server updates the OVO/Unix messages when it submits these messages as an AR ticket.

You define the submit rules by adding the ON_SUBMIT attribute to the target definition. You then assign keywords with parameters to the ON_SUBMIT attribute. Table 5-6 on page 110 shows the keywords you use to define for a target. All of the keywords are described in the sections that follow:

- Acknowledging Messages

- Taking Ownership of a Message (disowning first on request)

- Removing Ownership from a Message

- Modifying the text of a Message

- Generating an OVO message

- Adding an annotation

Table 5-6   **Submit Keywords**

Keyword	Parameters	Description
ACKNOWLEDGE	-	Use this keyword to acknowledge the OVO/Unix message right after the message is submitted as an AR ticket. This also removes the association between the action request and the OVO/Unix message from the OV SPI for Remedy database.
OWN	-	Use this keyword to take ownership of the OVO/Unix message right after the message is submitted as an AR ticket.
FORCE OWN	-	Use this keyword to remove the ownership of the message and then to take ownership of the OVO/Unix message right after the message is submitted as an AR ticket. Note that the REMSPI_ITO_ADMIN_PASSWD keyword is necessary in the remspi.cfg file for this to work.

Table 5-6   **Submit Keywords**

Keyword	Parameters	Description
DISOWN	-	Use this keyword to remove ownership of the OVO/Unix message right after the message is submitted as an AR ticket.
		Note that in order to use this keyword, the password for the OVO/Unix administrator has to be configured with the REMSPI_ITO_ADMIN_PASSWD keyword in the remspi.cfg file.
		Note further that this operation is carried out *after* the modification if the OVO/Unix message text (if specified).
MSGTXT_PREFIX	A *string* which will be prepended to the original message text. Note that besides ordinary text only the special variable ARS_ID is supported.	Use this keyword to modify the text of the OVO/Unix message right after the message is submitted as an AR ticket.
		Note that after the modification the message is automatically owned by the OVO/Unix core system. But you may *disown* the message right after this modification if you use the DISOWN submit keyword.
		Note further that an annotation documenting the modification is automatically added by the OVO/Unix core system.

Table 5-6   **Submit Keywords**

Keyword	Parameters	Description
MSG_IF_MANUAL	A *string* which is used as the message text. Note that besides ordinary text only the special variable ARS_ID is supported.	Use this keyword to specify the text of the OVO/Unix message which is generated right after the current message is successfully and manually submitted as an AR ticket ($ITO_SUBMITTER$ is not "OpC"). The new OVO message has the following attributes: • `Application "Remedy SPI"` • `Object "RemSPI"` • `Message Group "Remedy-SPI"` • `Severity "Normal"`
ANNO_IF_MANUAL	A *string* which is used as the annotation text. Note that besides ordinary text only the special variable ARS_ID is supported.	Use this keyword to specify the text of the annotation which is attached to the OVO/Unix message right after the message is submitted as an AR ticket. The annotation is added only for successfully and manually submitted messages. If multiple messages are submitted, the annotation is added to all messages.

Acknowledging Messages

The following example shows a submit rule that acknowledges the OVO/Unix message after the message is submitted as an AR ticket.

```
ON_SUBMIT
   ACKNOWLEDGE
```

For information on the problems that can occur when acknowledging a

---

large number of messages at the same time, see "Troubleshooting HP OV SPI for Remedy Usage" on page 178

Taking Ownership of a Message

The following example shows a submit rule that uses the FORCE OWN keyword.

The rule instructs the OV SPI for Remedy to take ownership of the OVO/Unix message (regardless which user owns the message currently) if the message is submitted as an AR ticket. If you omit the FORCE keyword, the message will be owned only there is no other user currently owning it.

```
ON_SUBMIT
   FORCE OWN
```

NOTE   The OVO/Unix user that you specify in the configuration file remspi.cfg becomes the owner of the OVO/Unix messages.

NOTE   If you use the FORCE keyword, you have to configure the password of the OVO/Unix administrator in the remspi.cfgconfiguration file using the REMSPI_ITO_ADMIN_PASSWD keyword.

For information on the problems that can occur when taking ownership of a large number of messages at the same time, see "Troubleshooting HP OV SPI for Remedy Usage" on page 178

Removing Ownership from a Message

The following example shows a submit rule that uses the DISOWN keyword. The rule instructs the OV SPI for Remedy to remove ownership from the corresponding OVO/Unix message if the message is submitted as an AR ticket.

```
ON_SUBMIT
   DISOWN
```

NOTE

If you use the FORCE keyword, you have to configure the password of the OVO/Unix administrator in the remspi.cfgconfiguration file using the REMSPI_ITO_ADMIN_PASSWD keyword.

Modifying the text of a Message

The following example shows a submit rule that uses the MSGTXT_PREFIX keyword. The rule instructs the OV SPI for Remedy to prepend the ticket of the constructed AR ticket to the text of the corresponding OVO/Unix message if the message is submitted as an AR ticket.

```
ON_SUBMIT
   MSGTXT_PREFIX "ID: $ARS_ID$"
```

NOTE

If you use the MSGTXT_PREFIX keyword, the message will be automatically get an annotation and will be automatically owned. You may decide to *disown* the message using the submit keyword DISOWN additionally.

Generating an OVO message

The following example shows a submit rule that uses the MSG_IF_MANUAL keyword. The rule instructs the OV SPI for Remedy to generate an OVO message if a message is successfully and manually submitted as an AR ticket. The submitter must not be "OpC".

NOTE

In the message text, the only action variable allowed is ARS_ID, which is substituted with the ID of the generated AR ticket.

```
ON_SUBMIT
   MSG_IF_MANUAL "AR ticket created with ID $ARS_ID$"
```

Adding an annotation

The following example shows a submit rule that uses the

ANNO_IF_MANUAL keyword. The rule instructs the OV SPI for Remedy to add an annotation to the OVO message which is submitted as an AR ticket. The annotation is sent only if the ticket was successfully and manually submitted. If multiple messages are submitted, the annotation is added to all of them.

NOTE   In the annotation text, the only action variable allowed is ARS_ID, which is substituted with the ID of the generated AR ticket.

```
ON_SUBMIT
    ANNO_IF_MANUAL "AR ticket created with ID $ARS_ID$"
```

Defining Conditions

Conditions filter the messages that OVO/Unix sends to the OV SPI for Remedy to determine whether to start an action or suppress the message.

- Message conditions determine whether to start an action and begin under MSGCONDITIONS.

- Suppress conditions determine whether to ignore a OVO/Unix message and begin under SUPPRESSCONDITIONS.

The rules file can contain multiple sets of message and suppress conditions in any order. You create each condition in a set by assigning a DESCRIPTION followed by CONDITION.

NOTE   The OV SPI for Remedy evaluates conditions in the order that they appear in the rules file. If a OVO/Unix message satisfies a condition, the OV SPI for Remedy does not evaluate any subsequent conditions. Consequently, you must consider the order of conditions in the rules file carefully.

Optional keywords and parameters are listed in brackets, e.g. [NOT] [LIKE]. The keyword NOT negates the specified check, the keyword LIKE switches from lexical comparison to pattern matching.

You define the condition using keywords and parameters that refer to

values in the OVO/Unix message.

NOTE

There is an exception from the last statement. The keyword EXTERNAL in the test part of a *condition* will refer to an *external value file*, not to a message attribute.

Table 5-7

Condition Keywords

Keyword	Description
AA_STATUS	The status of the automatic action of the OVO message. <u>Syntax</u>: `AA_STATUS [NOT] status` where `status` one of UNDEFINED, AVAILABLE, STARTED, FINISHED, FAILED
APPLICATION	The application string attached to the OVO message. <u>Syntax</u>: `APPLICATION [NOT] [LIKE] string`
CMA	This keyword needs two strings as parameters. The first parameter is the name of an OVO/Unix *Custom Message Attribute*, the second parameter the expected value of this *CMA*. This condition term is true if the OVO/Unix message has a Custom Message Attribute with that name and value. <u>Syntax</u>: `CMA key-string [NOT] [LIKE] val-string`
DUPLICATES	The number of duplicates of the OVO message. <u>Syntax</u>: `DUPLICATES op number` where `op` one of ==, !=, >=, <=, >, <

Table 5-7   **Condition Keywords**

Keyword	Description
EXTERNAL	An value in an external value file. For more details, see "Referencing External Values" on page 168.
	Note that this is keyword behaves different than others in this list. This keyword does not refer to an attribute of the OVO/Unix message, but to a value in an external file.
	Syntax: `EXTERNAL key-string [NOT] [LIKE] val-str`
ITO_SUBMITTER	The name of the OVO user who submitted the message.
	Syntax: `ITO_SUBMITTER [NOT] [LIKE] string`
MSGGRP	The message's message group.
	Syntax: `MSGGRP [NOT] [LIKE] string`
NODE	The name of the machine that sent the message. (or for which the message *was sent*).
	Syntax: `NODE [NOT] [LIKE] string`
NODE_GROUP	One of the OVO node groups the message node is a member of. Only one node group can be listed here. Note that there is no corresponding action variable `$NODE_GROUP$`.
	Syntax: `NODE_GROUP [NOT] string`
NODE_IP	The IP address of the machine that sent the message (or for which the message was *sent*).
	Syntax: `NODE_IP [NOT] [LIKE] string`

Table 5-7   **Condition Keywords**

Keyword	Description
NODE IN	A file is specified as the parameter. The current OVO message node is checked for existence in this file and the condition is satisfied if the node is a member of the file. The file holds only names or IP addresses in decimal form (a.b.c.d), one per line, and is plain text. Syntax: `NODE [NOT] IN string`
OA_STATUS	The status of the operator initiated action of this OVO message. Syntax: `OA_STATUS [NOT] status` where `status` one of `UNDEFINED, AVAILABLE, STARTED, FINISHED, FAILED`
OBJECT	The name of the object that generated the message. Syntax: `OBJECT [NOT] [LIKE] string`

OWN_USER	The name of the OVO user who owns the OVO message.
	
	Syntax:
	`OWN_USER [NOT] [LIKE] string`
	`OWN_USER [NOT] NOBODY`

SERVICE_NAME	The name of the service attached to the OVO message. Syntax: `SERVICE_NAME [NOT] [LIKE] string`
SEVERITY	The message's severity. This keyword can take one or more of the parameters unknown, normal, warning, minor, major, or critical. A list is treated as a logical OR. Syntax: `SEVERITY [NOT] severity` where severity is a blank separated list of `Unknown, Normal, Warning, Minor, Major, Critical`

Table 5-7   **Condition Keywords**

Keyword	Description
TEXT	The text that explains the message's cause, i.e. the message text. <u>Syntax</u>: `TEXT [NOT] [LIKE] string`

Table 5-7 shows the keywords that you can assign to a condition. All of the keywords are optional.

You must give a value for each keyword. If the values in the OVO/Unix message match the values you give for the keywords, the OVO/Unix message satisfies the condition, and the action starts.

NOTE   The exception is the keyword EXTERNAL. This keyword does refer to the values in the external file, it does *not* refer to the attributes of the OVO/Unix message.

The next example shows a suppress condition that filters out all OVO/Unix messages that have the severity Normal. From the OVO/Unix messages that remain, a message condition selects those that:

• belong to the message group Printers

• have the message text Paper Jam

• have the severity Critical, Major, or Minor

```
SUPPRESSCONDITIONS
  DESCRIPTION "Suppress Normal"
  CONDITION
    SEVERITY Normal

MSGCONDITIONS
  DESCRIPTION "ProbSys Condition 1"
  CONDITION
    MSGGRP "Printers"
    TEXT "Paper Jam"
    SEVERITY Critical Major Minor
```

NOTE

If you change the DESCRIPTION for a message condition that the OV SPI for Remedy has already used, the OV SPI for Remedy cannot update existing action requests for that condition.

Using Regular Expressions

This sections describes how regular expressions may be used as parameters to the condition keywords.

If the keyword LIKE is specified in a condition, the message attribute (or the external value) is not compared character by character with the string parameter following the LIKE keyword, but this string parameter is then treated as a regular expression pattern.

The SPI supports *Extended Regular Expressions*, like well known commands as grep, vi or awk.

NOTE

For a detailed discussion of *Extended Regular Expressions*, see the appropriate on-line operating system documentation on your OVO management server (HP-UX: man 5 regexp, Solaris: man 5 regex).

Caution: If you have to use the backslash symbol "\" in you regular expression pattern, you will have to quote this symbol itself by prepending an additional "\" symbol. This is a difference to the operating system documentation.

Table on page 120 lists some examples for using regular expressions in conditions.

Table 5-8

Syntax	Description
OBJECT LIKE "^oracle-[0-9]+$"	Matches if the OVO message object attribute starts with the string oracle-, and ends with a non empty sequence of digits.

Table 5-8

Syntax	Description
TEXT LIKE "ERROR|WARNING"	Matches if the OVO message text contains the words ERROR or WARNING (or both).
NODE_IP LIKE "10\\.1\\.2\\.25[0-4]"	Matches if the message node's IP address is in the range from 10.1.2.250 to 10.1.2.254. Note that you have to quote the period, since a period matches any single character. More, for the SPI it's necessary to quote the quote character, too.

Defining Actions

This section describes how to define actions within the HP OV SPI for Remedy. The section covers the following areas:

- Using Action Variables
- Defining Values using Composite Formats
- Setting Default Values
- Setting Enumerated Field Values
- Handling Multiple OVO/Unix Messages
- Setting Update Policies
- Setting a Target's Default Actions
- Configuring automatic attachments

Each condition under MSGCONDITIONS has one or more actions. An action determines how the OV SPI for Remedy creates or updates an action request when a OVO/Unix message satisfies the condition. To define an action, you specify the target and state how the OV SPI for Remedy must create or update fields.

To specify a target, use SET TARGET *target name*. To specify a field, use FIELD *field ID value*. To specify further actions, repeat the TARGET

section further.

The next example adds two actions to an existing condition definition. The first action places the text Created by SPI into field 1000000005 in the schema that the target ProbSys1 defines. The second action places text into a field in a schema that the target ProbSys2 defines.

```
MSGCONDITIONS
  DESCRIPTION "ProbSys Condition 1"
  CONDITION
    MSGGRP "Printers"
    TEXT "Paper Jam"
    SEVERITY Critical Major Minor

  SET
   TARGET "ProbSys1"
     FIELD 1000000005 "Created by SPI."

   TARGET "ProbSys2"
     FIELD 1000000005 "Copy of AR on ProbSys1. Created by SPI."
```

NOTE   Remember to ensure that the value you *assign* to a field is of the appropriate *type* for the field. For example, if the schema defines a field as an integer field, you should assign integers to it.

Even if you do *not* do this, the OV SPI for Remedy nonetheless attempts to convert in the appropriate manner. The OV SPI for Remedy truncates string values if necessary.

Using Action Variables

You can also define values in the action request using information derived from the OVO/Unix message.

Table 5-9 shows the action variables that you can use. Each time you use

an action variable, place $ before and after the variable name.

Table 5-9   **Action Variables**

Variable	Description	Can Be Empty?
AA_ACK_FLAG	Whether this message's automatic action acknowledges the message.	No
AA_ANNO_FLAG	Whether the output of this message's automatic action becomes a new annotation.	No
AA_CMD	The command for this message's automatic action.	Yes
AA_NODE	The node which this message's automatic action runs on.	Yes
AA_STATUS	The status of this message's automatic action.	No
ACK_TIME	The time that the message became acknowledged.	Yes
ACK_USER	The name of the OVO/Unix user who acknowledged the message.	Yes
ANNO_CNT	The number of annotations to this message.	No
ANNO_USER	The name of the OVO/Unix user who added an annotation last.	Yes
DUPLICATES	The number of duplicates of this OVO message.	No
ESCAL_SRV	The name of the server from or to which the message is escalated.	Yes
ESCAL_STAT	Whether the message is escalated to or from another management server, or not at all.	No

Table 5-9   **Action Variables**

Variable	Description	Can Be Empty?
EXTERNAL	This action variable needs the name of a key in the *external file* as a parameter. This pair is then replaced by the *value* assigned to the *key* in the *external value* file. The following example will insert into AR field 8 the value of the key `PhoneNumber` in the external value file: `FIELD 8 "Phone = $EXTERNAL PhoneNumber$"`	Yes
HLP_TEXT	The instructions associated with the message. If OVO/Unix has an external instruction interface for this message, the OV SPI for Remedy uses the resolved instruction *only* if the message was submitted automatically (via the Trouble Ticket Interface) , otherwise the interface name and command is used as the value of this keyword. Note that in order to be able to fetch the resolved instruction text, the *Instruction Text Interface* must be configured accordingly (Resolved text for Trouble Ticket Interface)	Yes
HOST_NAME	The host name of the physical machine where the OVO/Unix management server and OV SPI for Remedy server run. See *MANAGEMENT_SERVER* for the name of the current OVO management server/package when used in cluster environments.	No

Table 5-9   **Action Variables**

Variable	Description	Can Be Empty?
INSTR_IF	The name of any instruction interface for this message.	Yes
INSTR_PAR	Details of any instruction interface parameters.	Yes
INSTR_TYPE	Whether an instruction for this message is internal or uses an interface.	Yes
ITO_MSI	Whether OVO/Unix passed the message to the message stream interface.	No
ITO_NOTIFY	Whether OVO/Unix passed the message to a notification interface.	No
ITO_SUBMITTER	The name of the OVO user who submitted the messages. If the messages was submitted automatically (Trouble Ticket Interface), this value is "OpC".	No
ITO_TT	Whether OVO/Unix passed the message to the trouble ticket interface.	No
ITO_TT_ACK	Whether OVO/Unix acknowledges this message automatically after it passes it to the trouble ticket interface.	No
ITO_UNMATCHED	Whether the message did not match a condition on the OVO/Unix agent.	No
ITO_USER	The OVO/Unix user name which the OV SPI for Remedy used to log in to OVO.	No
LOG_ONLY	Whether OVO/Unix immediately moves the message to the history database.	No

Table 5-9   **Action Variables**

Variable	Description	Can Be Empty?
MSG_ANNO	The annotations added to the message. When the OV SPI for Remedy creates an action request, it combines all the existing annotations. When the OV SPI for Remedy updates an action request, it uses only the last annotation.	Yes
MSG_GRP	The message's message group.	No
MSG_ID	The messages ID.	No
MSG_OBJ	The name of the object that generated the message.	No
MSG_SEVERITY	The message's severity.	No
MSG_SRC	The message's source.	No
MSG_SRC_TYPE	The message's source type.	No
MSG_STATUS	Whether the message is active or in the history database.	No
MSG_TEXT	The text that explains the message's cause.	No
MSG_TEXT_ORIG	The message's original text before any changes.	No
MSG_TYPE	The message's type.	Yes
NODE_APPL	The application that generated the message.	No
NODE_DATE	The date on which the OVO/Unix managed node generated the message.	No
NODE_NAME	The name of the machine that sent the message.	No

Table 5-9 **Action Variables**

Variable	Description	Can Be Empty?
NODE_IP	The IP address of the machine that sent the message (in decimal string format, e.g. 192.168.0.12)	No
NODE_TIME	The time at which the OVO/Unix managed node generated the message.	No
OA_ACK_FLAG	Whether this message's operator initiated action acknowledges the message.	No
OA_ANNO_FLAG	Whether the output of this message's operator initiated action becomes a new annotation.	No
OA_CMD	The command for this message's operator initiated action.	Yes
OA_NODE	The node which this message's operator initiated action runs on.	Yes
OA_STATUS	The status of this message's operator initiated action.	No
OWN_TIME	The time at which a user took ownership of this message.	Yes
OWN_USER	The name of the OVO/Unix user who owns the message.	Yes
SERVICE_NAME	The name of the service attached to the OVO message.	Yes
SRV_DATE	The date on which the OVO/Unix management server received the message.	No
SRV_TIME	The time at which the OVO/Unix management server received the message.	No
MANAGEMENT_SER VER	The name of the current OVO management server system or OVO package (in cluster environments).	No

Table 5-9 shows the action variables that you can use.

NOTE   Each time you use an action variable, place $ before and after the variable name.

The next example shows an action definition that uses variables to define the values in an action request.

```
SET TARGET "ProbSys"
    FIELD 1000000003 "$NODE_NAME$"
    FIELD 1000000004 "$NODE_APPL$"
    FIELD 1000000005 "Message created at $NODE_TIME$ on
$NODE_DATE$\\nMessage received at $SRV_TIME$ on $SRV_DATE$"
```

The action puts:

• the name of the managed node that created the OVO/Unix message into field `1000000003`

• the name of the application that generated the OVO/Unix message into field `1000000004`

• a message that contains times and dates of creation and receipt into field `1000000005`

NOTE   Some of the action variables can be empty if there is no appropriate value for them. You may need to define defaults when using these variables. For more details, see "Setting Default Values" on page 130.

Defining Values using Composite Formats

You can also define values in the action request using pre-formatted information from the OVO/Unix message. You do this using composite formats. Table 5-10 shows the composite formats available. The composite format `MSG_EVENT` varies automatically, depending on how the OVO/Unix message changed. Table 5-11 shows the formats that can

apply for MSG_EVENT.

Table 5-10   **Composite Formats**

Format Name	Description
MSG_EVENT	A format that varies automatically, depending on the type of change made to the OVO/Unix message. Table 5-11 shows a list of possible formats.
MSG_PARMS	A format that shows common message parameters as follows: "Message: MSG_TEXT Message ID: MSG_ID Severity: $MSG_SEVERITY$ Instructions: HLP_TEXT"

Table 5-11   **MSG_EVENT Formats**

Event	Format
MSG_EVT_AA_END	"Automatic action of OVO/Unix message MSG_ID finished."
MSG_EVT_AA_START	"Automatic action of OVO/Unix message MSG_IDhas been started."
MSG_EVT_ACK	"ITO message MSG_ID has been acknowledged by ACK_USER."
MSG_EVT_ANNO	"ITO message MSG_ID has been annotated by $ANNO_USER$."
MSG_EVT_CREATE	"ITO message MSG_ID has been submitted to the TroubleTicket IF."
MSG_EVT_DISOWN	"ITO message MSG_ID has been disowned."
MSG_EVT_ESC_FROM	"ITO message MSG_ID has been escalated from $ESCAL_SRV$."
MSG_EVT_ESC_TO	"ITO message MSG_ID has been escalated to $ESCAL_SRV$."

Table 5-11   **MSG_EVENT Formats**

Event	Format
MSG_EVT_OA_END	"Operator-initiated action of ITO message MSG_ID finished.'
MSG_EVT_OA_START	"Operator-initiated action of ITO message MSG_ID has been started."
MSG_EVT_OWN	"ITO message MSG_ID has been owned by OWN_USER."
MSG_EVT_UNACK	"ITO message MSG_ID has been unacknowledged."

Setting Default Values

Some of the action variables in Table 5-9 may contain no value, depending on the OVO/Unix message. For example, if no user owns the message the variable OWN_USER is empty. Submission of an action request fails if your action tries to place an empty variable into a mandatory schema field.

To avoid this problem, you can specify a default value for a field. The following example shows an example of a default value. If the variable OWN_USER is empty, the action puts the text Not owned into field 1000000005 instead.

FIELD 1000000005 "Owned by OWN_USER." "Not owned."

NOTE   The default value for an enumerated field is the first value. You cannot specify a different default value for an enumerated field. For more details, see "Setting Enumerated Field Values" on page 130.

Setting Enumerated Field Values

Enumerated fields are fields that can have one of a restricted number of values. In a schema, an enumerated field is either a drop-down list or a radio button field. Figure 5-4 shows example enumerated fields in a schema.

Figure 5-4   **Enumerated Fields**

To set values for enumerated fields in a schema, you use text or a variable that matches one of the available values.

You can also use a variable whose values from the OVO/Unix message do not match the available values in the schema. To do this, you map the index numbers of the OVO/Unix message values to the index numbers of the schema values.

The variable *MSG_SEVERITY* can have the value unknown, normal, warning, minor, major, or critical. These have index values from 0 to 5 respectively. The field Impact in Figure 5-4 can have the value critical, high, warning, normal, or unknown. These have index values from 0 to 4 respectively.

The following example shows how to map the variable *MSG_SEVERITY* to the field Impact by listing the field's index numbers in the order they should be used.

```
FIELD 1000000002 "$MSG_SEVERITY$" "4, 3, 2, 1, 1, 0"
```

The example maps:

- unknown in OVO/Unix (0) to unknown (4)

- normal in OVO/Unix (1) to normal (3)

- warning in OVO/Unix (2) to warning (2)

- minor in OVO/Unix (3) to high (1)

- major in OVO/Unix (4) to high (1)

- critical in OVO/Unix (5) to critical (0)

Handling Multiple OVO/Unix Messages

You can submit multiple OVO/Unix messages to create just one action

request. When you do this, the OV SPI for Remedy has a number of values available for every variable you use in the action definition. To specify which value the OV SPI for Remedy uses, insert a multiple message keyword before the FIELD keyword.

Table 5-12 **Multiple Message Keywords**

Keyword	Description
ALL	For numeric fields, the OV SPI for Remedy uses the highest value from all the OVO/Unix messages.
	For text fields creates a numbered list that includes values from all the OVO/Unix messages.
	This is the default for text fields.
FIRST	The OV SPI for Remedy uses the value from the first OVO/Unix message in order of severity.
	This is the default for numeric and enumerated fields.

Table 5-12 shows the multiple message keywords that you can use. The example below shows multiple message policies where:

- field 1000000002 contains the severity of the first message in a group (which is the severest)

- field 1000000001 contains a numbered list of annotations from all the messages in a group

```
FIRST FIELD 1000000002 "$MSG_SEVERITY$"
ALL FIELD 1000000001 "$MSG_ANNO"
```

Setting Update Policies

By default, the OV SPI for Remedy does not update any fields in the action request if the OVO/Unix message changes. To specify that the OV SPI for Remedy should update a field when necessary, insert the UPDATE keyword before the FIELD keyword. You can extend this with more

specific update keywords.

Table 5-13   **Update Keywords**

Keyword	Description
ATTACHED	Optional keyword which controls the handling of the original OVO message in case of attaching more OVO messages to an existing AR ticket. May be used with any of the following keywords: APPEND, OVERWRITE. This keyword is ignored for automatic attachments and submit requests.
	If ATTACHED is specified for a field, the SPI ignores the value of the original OVO message when determining the new value of the field and only considers the *attached* OVO messages.
APPEND	For numeric fields, the OV SPI for Remedy replaces the existing value only if the new value is higher.
	For text fields, the OV SPI for Remedy adds the new value after the existing value.
	This is the default for text fields.
OVERWRITE	The OV SPI for Remedy always replaces the existing value with the new value.
	This is the default for numeric fields.

Table 5-13 shows the different update keywords you can use. The example below shows examples of update policies where the OV SPI for Remedy:

- appends annotations into field 1000000001; if new OVO messages are attached to an existing ticket, the annotation of the original OVO message (which created the ticket) is ignored.

- overwrites field 1000000007 with details of the latest event

```
UPDATE ATTACHED FIELD 1000000001 "$MSG_ANNO$"
UPDATE OVERWRITE FIELD 1000000007 "$MSG_EVENT$"
```

NOTE   The OV SPI for Remedy always appends values to diary fields in a schema.

Setting a Target's Default Actions

You can set the default actions for a target by adding them to the target definition. The OV SPI for Remedy uses the default actions every time the SET TARGET keywords reference that target.

You can override the default actions when you define an action that belongs to a specific condition.

The following example shows a target definition with default actions at the end. In this example, every time an action specifies the target ProbSys, the OV SPI for Remedy enters:

- OVO Smart Plug-In in field 2

- New in field 7

- the message text in field 8

```
TARGET "ProbSys"
  SERVER "ars1.bbn.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "C338D5F22121EO76C200F6E6B4E94B2C679B2DA"
  ITO_UPDATE
    ANNOTATE "Work log entry:\\n$1000000001$"

  FIELD 2 'OVO Smart Plug-In"
  FIELD 7 "New"
  FIELD 8 "$MSG_TEXT$"
```

NOTE   You can also include action keywords within the default actions defined for a target. This includes action variables, composite formats, default field values, enumerated fields, multiple message policies, and update policies.

Configuring automatic attachments

Up to now, OVO messages could only be *submitted* both automatically and manually. But *attaching* new OVO messages to an existing AR ticket could only be done manually.

With the new ATTACH_TO keyword, it's possible to define a list of *search fields* which are then used to locate an AR ticket, to which the *submitted*

OVO messages should be attached to.

For search fields, it is possible to specify a list of values the field may have to satify the condition, i.e. the values are OR'ed together.

If the condition matches, all AR servers listed as a SET TARGET of this condition are searched for tickets for which the fields contained in the search list have the specified values.

- If such tickets are found on a target, the submitted OVO messages are attached to the oldest ticket.

- If no tickets are found on a target, a new ticket is created as if the OVO messages were submitted without the search fields.

The search fields are specified after the last sub condition and before the SET TARGET's are defined.

The following example shows a condition definition with search fields. In this example, every time an OVO message is *submitted* (not attached) and matches this condition (severity is Critical, object matches "ora.*"), all tickets on target "ProbSys" are located which have their field 9 set to the actual object attribute of the submitted OVO message.

If such tickets are found, the currently submitted OVO messages are attached to the oldest of these tickets using the specified set actions (the message text is appended to field 10). If no such tickets are found, then a new ticket is created with field 9 set to the message object and field 10 set to the message text.

```
DESCRIPTION "Attach-1"
 CONDITION
  SEVERITY  Critical
  OBJECT    LIKE "ora.*"
  ATTACH_TO
   FIELD 9 "$MSG_OBJ$"
 SET
  TARGET "ProbSys"
   FIELD  9 "$MSG_OBJ$"
   UPDATE APPEND FIELD 10 "$MSG_TEXT$"
```

NOTE You can also include action keywords within the search fields defined for a condition. This includes action variables and enumerated fields.

It's possible to use multiple search fields and multiple SET TARGET's and

even multiple values for a search field. In the example listed below, field 7 may contain "Assigned" or "Accepted" in order for this search condition to be true. Thus, an attachment is made if a ticket is found with field 9 set to the value of the current message object **AND** field 7 set to "Assigned" **or** "Accepted".

```
ATTACH_TO
  FIELD  9 "$MSG_OBJ$"
  FIELD  7 "Assigned", "Accepted"
SET
 TARGET "ProbSys"
  FIELD  9 "$MSG_OBJ$"
  UPDATE APPEND FIELD 10 "$MSG_TEXT$"
 TARGET "OtherTarget"
  FIELD 11 "Something"
  FIELD  9 "$MSG_OBJ"
```

NOTE

If multiple OVO messages are submitted and action variables are used in the search field definition, then only the FIRST (according to the rules described earlier) OVO message is evaluated.

The search fields are only evaluated for message submittals, not for message attachments, i.e. the operation has to be a SUBMIT (regardless whether it has been started manually or automatically).

An Example of a Rules File

This section uses the rules file supplied with the OV SPI for Remedy and displayed in Example 5-1 to demonstrate how the field names of an ARS schema shown in Figure 5-5 on page 137, are associated with unique identifiers, shown in Table on page 137. This section covers the following topics:

- Submitting OVO/Unix Messages
- Updating an Action Request
- Updating a OVO/Unix Message

Example 5-1 illustrates how any changes to the fields you identify may then be linked to actions that need to be carried out in OVO/Unix. The AR schema shown in Figure 5-5 enables help desk users to create and

update details of reported problems.

Figure 5-5   **Example Schema**
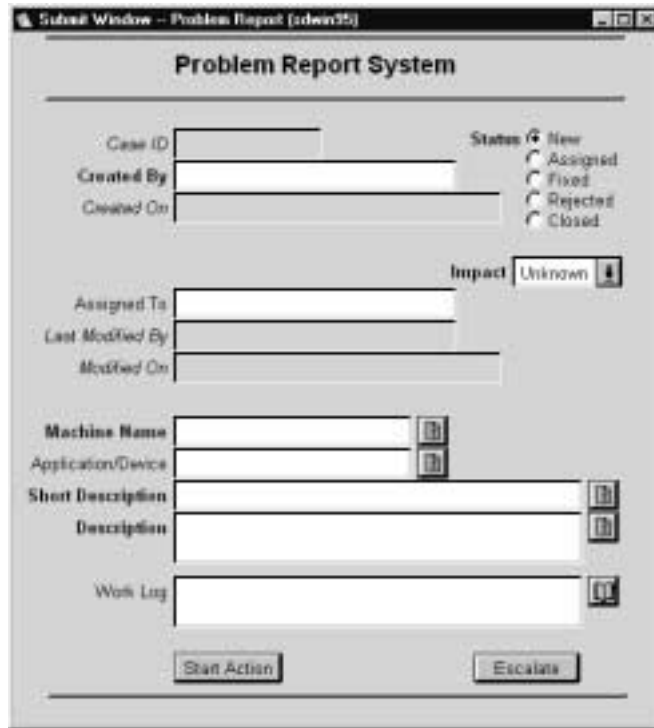
Table 5-14 shows the unique identifiers for each of the fields in the Problem Report System schema. The rules file uses the identifiers to reference the fields and their values.

Table 5-14   **Example Schema Field Identifiers**

Field Name	Identifier
Case ID	1
Created By	2
Created On	3
Assigned To	4
Last Modified By	5

Table 5-14   **Example Schema Field Identifiers**

Field Name	Identifier
Modified On	6
Status	7
Short Description	8
Work Log	1000000001
Impact	1000000002
Machine Name	1000000003
Application/Device	1000000004
Description	1000000005

Example 5-1 shows a simple rules file that defines how the OV SPI for Remedy handles OVO/Unix messages and action requests in the Problem Report System.

Example 5-1   **Example Rules File**

```
#*** File Definition ***

SYNTAX_VERSION 2.2
SPI_RULES "rules"
DESCRIPTION "Example rules for the Problem System"


#*** Target Definition ***

TARGET "ProbSys"
  SERVER "sdwin95.hp.com"
  SCHEMA "Problem Report"
  USER "spi-user"
  PASSWORD "C338D5F21219E076C200F6E6B4E946B2C679B2DA"

  #*** Update Rules ***

  ITO_UPDATE
    ANNOTATE
"Action request modified.
Schema: $ARS_SCHEMA$
```

```
Server: $ARS_SERVER$
Latest work log entry: $1000000001$"
    ANNOTATE IF_MODIFIED 1000000001
            "Work log modified: $1000000001$"
    ANNOTATE IF_MODIFIED 7 "Closed"
            "Action request closed by $5$ on $6$"

    OWN         IF_MODIFIED 7 "Assigned"
    OWN         IF_MODIFIED 7 "Fixed"
    ACKNOWLEDGE IF_MODIFIED 7 "Closed"
    OP_ACTION "operator-action"
    ESCALATE  "escalate"

  #*** Submit Rules ***

  OWN_SUBMIT
    DISOWN
    MSGTXT_PREFIX "AR-ID: $ARS_ID$"

  #*** Default Actions ***

  FIELD 2 "ITO Smart Plug-In"
  FIELD 7 "New"
  FIRST FIELD 8 "$MSG_TEXT$"
  FIRST FIELD 1000000002 "$MSG_SEVERITY$" "4, 3, 2, 1, 1, 0"
  FIELD 1000000003 "$NODE_NAME$"


#*** Suppress Conditions ***

SUPPRESSCONDITIONS

  DESCRIPTION "Suppress Normal"
  CONDITION
    SEVERITY Normal

#*** Message Conditions ***

MSGCONDITIONS

  DESCRIPTION "Condition1"
  CONDITION
    MSGGRP "Printers"
    SEVERITY Warning

  SET
```

```
  TARGET "ProbSys"
   FIELD 1000000004 "$MSG_OBJ$"
   FIELD 7 "Closed"
   ALL FIELD 1000000005
"Printing Error - INFORMATION ONLY - TAKE NO ACTION.
* This action request is linked to ITO message: $MSG_ID$
* ITO message text: $MSG_TEXT$
* ITO message generated at $NODE_TIME$ on $NODE_DATE$"



  DESCRIPTION "Condition2"
  CONDITION
    MSGGRP "Printers"
    SEVERITY Minor Major Critical

  SET
   TARGET "ProbSys"
    UPDATE ALL FIELD 1000000001 "$MSG_EVENT$"
    ALL FIELD 1000000004 "$MSG_OBJ$"
    ALL FIELD 1000000005
"Printing Error.
* This action request is linked to ITO message: $MSG_ID$
* ITO message text: $MSG_TEXT$
* ITO message generated at $NODE_TIME$ on $NODE_DATE$
* Instructions: $HLP_TEXT$"


#*** More Suppress Conditions ***

SUPPRESSCONDITIONS

  DESCRIPTION "Suppress Performance"
  CONDITION
    MSGGRP "Performance"


#*** More Message Conditions ***

MSGCONDITIONS

  DESCRIPTION "Condition3"
  CONDITION
    SEVERITY Critical

  SET
```

```
    TARGET "ProbSys"
     UPDATE ALL FIELD 1000000001 "$MSG_EVENT$"
     FIELD 1000000004 "$NODE_APPL$"
     ALL FIELD 1000000005
"Critical Error. Submitted from ITO.
$MSG_PARMS$
Annotation: $MSG_ANNO$"
```

Submitting OVO/Unix Messages

Figure 5-6 shows two OVO/Unix messages that appear when a large
printer fails.

Figure 5-6   **OVO/Unix Messages for Submission**
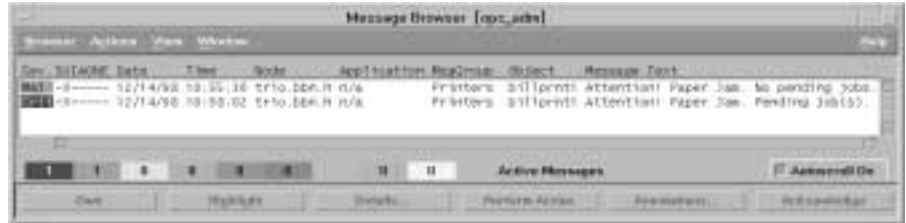
When the OVO/Unix user decides to submit these messages to the AR
System, and the following occurs:

1. The OV SPI for Remedy begins checking for a matching condition. It
   evaluates the second message against conditions, because this has
   the highest priority.

2. `Suppress Normal` in the rules file does not match, so the OV SPI for
   Remedy continues checking.

3. `Condition1` in the rules file does not match, so the OV SPI for
   Remedy continues checking.

4. `Condition2` in the rules file matches, so the OV SPI for Remedy
   checks no further.

5. The OV SPI for Remedy completes the default actions for the target
   `ProbSys`.

6. The OV SPI for Remedy completes the actions for `Condition2`.

7. The OV SPI for Remedy completes the actions listed in the
   `ON_SUBMIT`section.

Figure 5-7   **Created Action Request**
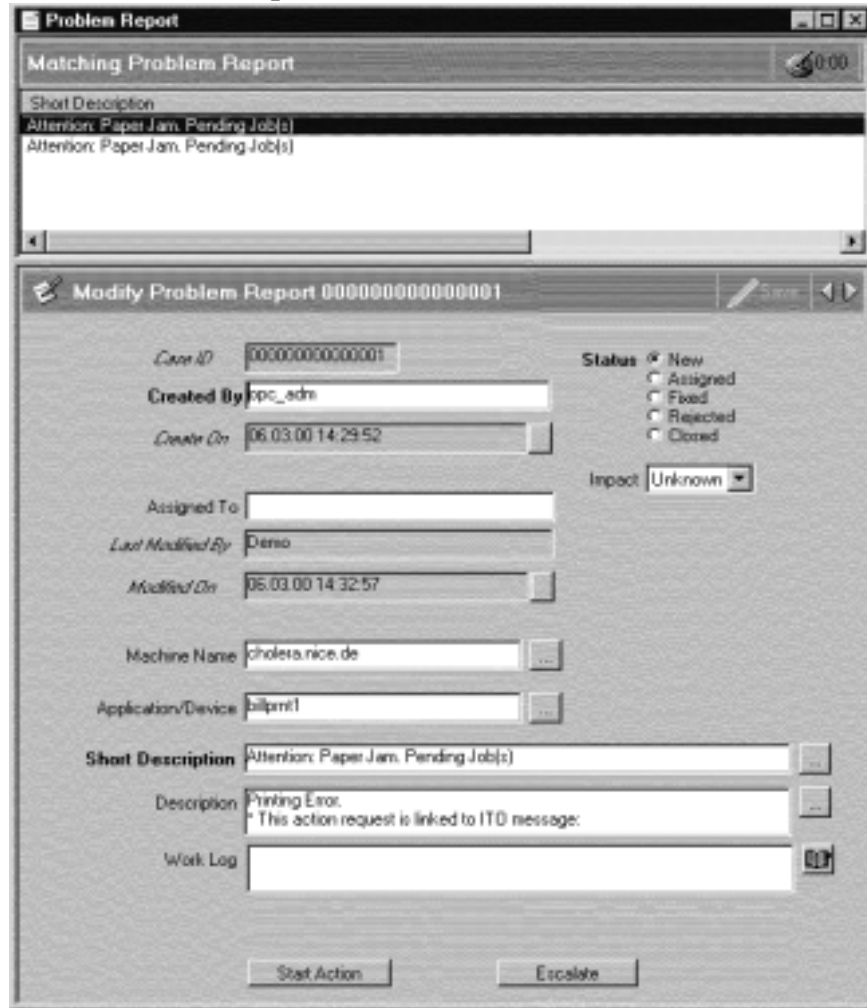
Figure 5-7 shows the action request that the OV SPI for Remedy creates
as a result of the OVO/Unix message. Notice that:

- the Impact field has the severity Critical

- the fields Machine Name, Application/Device, Description,
  and Work Log each contain two entries. The multiple message policy
  specifies one entry per OVO/Unix message.

NOTE   The first condition in Example 5-1 creates a different action request if the OVO/Unix message is within the `Printers` message group, but is only a warning. This involves overriding the target's default action for field `7`.

The third condition in Example 5-1 creates an action request for all critical OVO/Unix messages that do not meet the first or second conditions.

Updating an Action Request

A OVO/Unix user has experience with this type of large printer, and notices the OVO/Unix message in the message browser. She adds an annotation to the OVO/Unix message explaining that paper jams sometimes occur when a certain part wears out.

Because the OV SPI for Remedy applied the second condition in Example 5-1 when creating the action request, it also applies the second condition when updating it.

Figure 5-8   **Updated Action Request**

In the second condition, the OV SPI for Remedy can update the `Work Log` field with details of a message event. Therefore, when the OVO/Unix user adds the annotation, the OV SPI for Remedy appends details of this event to the action request's work log. Figure 5-8 shows the how the annotation in OVO/Unix has updated the work log entry in the action request.

NOTE   When the OV SPI for Remedy applies the first condition in Example 5-1, it cannot update any of the action request's fields.

When the OV SPI for Remedy applies the third condition in Example 5-1, it can only update the `Work Log` field.

Updating a OVO/Unix Message

Figure 5-9 shows how amendments to the `Status` and `Work Log` of the action request cause an annotation to be added automatically to the corresponding OVO/Unix message—the same OVO/Unix message, which created the Action Request being amended here. When the user saves the amendments, a filter calls the OV SPI for Remedy client, which sends details of the action request and any modifications to the OV SPI for Remedy server.

Figure 5-9 **Changed Action Request**
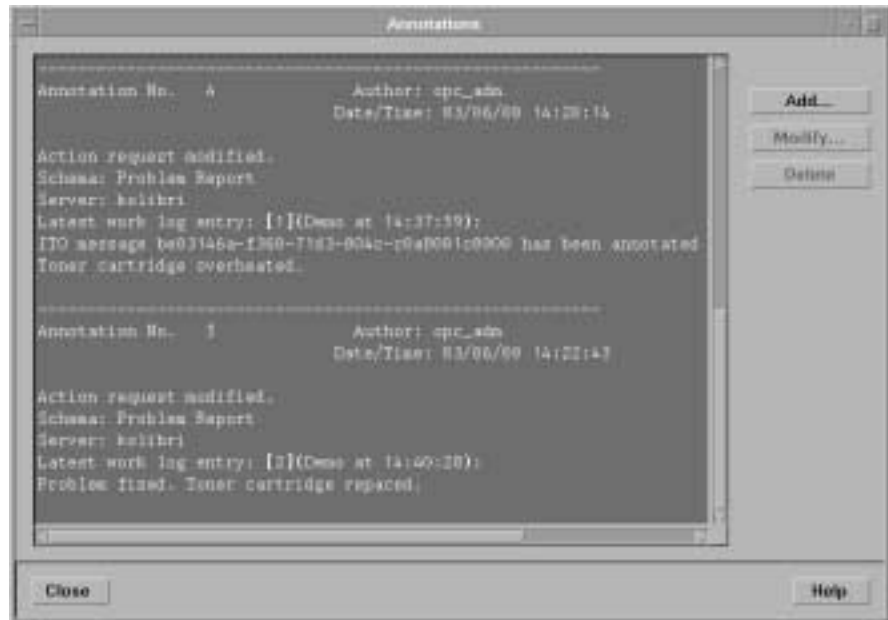
This target's update rules specify that the OV SPI for Remedy must then:

- add an annotation to the OVO/Unix message that includes the Work Log entry

- take ownership of the OVO/Unix message because the Status is now Assigned

NOTE The update rules apply for all action requests that belong to a schema, provided that the OV SPI for Remedy created them.

ARS Filters and Active Links

To enable ARS to communicate with OVO/Unix via the OV SPI for Remedy you need to complete the following steps.

1. Configure one or more ARS filters that run when somebody modifies or deletes an action request

2. Configure active links in ARS which enable users to start actions or escalate messages in OVO/Unix directly from the AR schema

This section describes the following topics:

- The itoupdate Command

- The remspifilter Command

- Configuring the ARS Filter Automatically

- Configuring the ARS Filter Manually

- Configuring an Active Link

NOTE   You should consider to create the AR filters manually if you want to notify the Remedy SPI server only if an interactive AR user has changed the ticket. The filters constructed automatically by `remspifilter` execute on any modification of the ticket, regardless whether it was an interactive AR user or the Remedy SPI server itself. This can lead to multiple annotations to the OVO/Unix message.

The itoupdate Command

The AR filters and active links call the OV SPI for Remedy client. The OV SPI for Remedy client is the `itoupdate` utility, which is copied to the AR server either during the distribution of the OV SPI for Remedy templates or manually with the `ftp` command. For more information about template assignment and distribution, see "Installing the OV SPI for Remedy" on page 44.

The `itoupdate` utility is installed on the AR server system in either one of the following locations, depending on the operating system:

UNIX	`/var/opt/OV/bin/OpC/cmds/itoupdate`
Windows NT	`\usr\OV\bin\OpC\cmds\itoupdat.exe`

The `itoupdate` command informs the HP OV SPI for Remedy server process about changes made to the AR ticket. In addition, `itoupdate` is used by the ARS `User Tool` to trigger actions attached to OVO/Unix messages, for example; the execution of operator-initiated actions, escalating a message, and so on.

The `itoupdate` command may be used manually and accepts the following options:

`itoupdate <Remedy SPI server hostname> <options> ...`

where the permitted options are:

ENTRY_ID	the ARS ticket id
SERVER	the name of the ARS server
SCHEMA	the name of the ARS schema
CMD	a command name as defined in the `remspi.cfg` file
OP	the operation you wish to perform on an ARS ticket.
MODIFIED	the ID of the field which was modified in the current transaction

Examples of usage:

1. To use the `itoupdate` command to trigger an action on a OVO/Unix message:

   itoupdate *<Remedy SPI server>* \

       ENTRY_ID=*<AR ticket id>* \

       SERVER=*<AR server name>* \

       SCHEMA=*<AR schema name>*\

       CMD=*<command name as defined in remspi.cfg>*

2. To use the `itoupdate` command to inform OVO/Unix about changes to an AR ticket:

   itoupdate *<Remedy SPI server>* \

       ENTRY_ID=*<AR ticket id>* \

       SERVER=*<AR server name>* \

```
SCHEMA=<AR schema name> \

OP=<operation on AR ticket>
```

Note that the value of the `OP=` parameter may be taken directly from AR by using the AR variable *$OPERATION$*.

3. To use the `itoupdate` command to inform OVO/Unix about a changed field in an AR ticket:

```
itoupdate <Remedy SPI server> \

    ENTRY_ID=<AR ticket id> \

    SERVER=<AR server name> \

    SCHEMA=<AR schema name> \

    OP=<operation on AR ticket> \

    MODIFIED=<ID of modified field>
```

Note that the value of the `OP=` parameter may be taken directly from AR by using the AR variable *$OPERATION$*. The value of the `MODIFIED=` parameter is the numeric ID of the field.

4. To use the `itoupdate` command move an AR ticket from one (source) AR server to another (destination) AR server:

```
itoupdate <RemedySPI server> ENTRY_ID=<AR source ticket
id>\ SERVER=<AR source server name> SCHEMA=<AR source
schema\ name> CMD=DSO-TRANSFER ENTRY_ID=<AR destination
ticket id>\ SERVER=<AR destination server name>\
SCHEMA=<AR_destination_schema name>
```

See "Moving an AR ticket between AR servers" on page 157 for details.

The remspifilter Command

The `remspifilter` command registers the filters you create in ARS and want to use to perform actions in OVO/Unix. The `remspifilter` command accepts the following options:

```
remspifilter  [ -help | -version ]

            [ -create | -delete ]

            [ -target <target> |

            -server <server> -schema <schema> -user <user>
```

```
[ -passwd <password> ] ]
[ -ar_nt | -ar_ux ]
[ -cmd <filter command> ]
[ <filter name> ]
```

Where:

-ar_nt	indicates an AR server running on Windows NT
-ar_ux	indicates an AR server running on UNIX
-create	creates a new filter. Omit this option if you want to retrieve and list an existing filter.
-cmd	• *path of itoupdate* enables the created filter to find the OV SPI for Remedy client. Enter the path to the location where you installed the OV SPI for Remedy client. If you omit this option, the command uses the default HP-UX 10.x/11/x path; `/var/opt/OV/bin/OpC/cmds/` • *itoupdate* retrieves details of existing filters
-delete	deletes the filter you specify in: *filter name*.
filter name	defines the name of the filter to create, delete, or retrieve. The default *filter name* is: *<Schema Name>*:ITO-Update
-passwd	defines a password to be associated with the name defined in -user
-server	defines the name of the AR server. Note that inconsistencies between the use of short and long hostnames here and elsewhere can lead to problems.
-schema	defines the name of the schema
-target	creates a filter for a *target* defined in the rules file. Note that differences between target definitions and filter declarations are resolved on the HP OV SPI for Remedy server.
-user	defines the user name with which to log in to the AR

server

NOTE

The remspifilter command is not able to create the special filters needed if the IF_MODIFIED tag is used in the rules file. These filters must be created manually, see pConfiguring a special IF_MODIFIED ARS Filter Manually on page page 152.

Configuring the ARS Filter Automatically

To automatically configure an AR System filter, use the command remspifilter. For example, you could enter the following command:

```
remspifilter -create
-cmd /Program_Files/OpenView/Remspi/itoupdate -server ars1
-schema HD:HelpDesk -user admin
```

If you want to create filters for all the targets defined in the rules file, use only the -create option and the -cmd option if necessary. For example, you could enter the following command:

```
remspifilter -create
-cmd /Program_Files/OpenView/Remspi/itoupdate
```

Configuring the ARS Filter Manually

You can configure the filter manually in the same way that you create other filters in an AR System. To do this:

1. Create a new filter that executes when someone modifies or deletes an action request as shown in Figure 5-10.

Figure 5-10   A New Filter
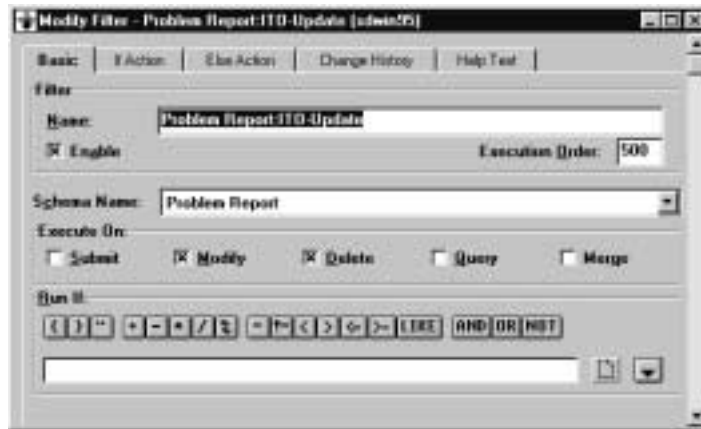
2. Add a run-process action that calls the OV SPI for Remedy client as shown in Figure 5-11.

Figure 5-11   A Filter's Run Process
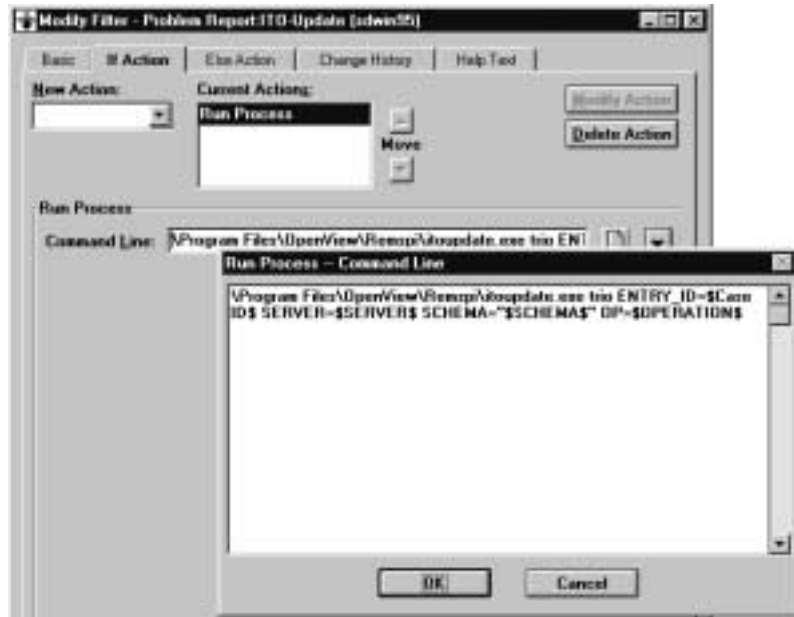
Configuring a special *IF_MODIFIED* ARS Filter Manually

This special filter is used to notify the Remedy SPI server that a certain field in an AR ticket has been changed by an interactive user. This special notification then triggers the evaluation of update rules which have the IF_MODIFIED tag set for this field in the rules file.

NOTE   Note that you have to configure one of these filters for every field which is used with the IF_MODIFIED tag in the rules file.

You can configure the filter manually in the same way that you create other filters in an AR System. To do this:

1. Create a new filter that executes when someone modifies or deletes an action request as shown in Figure 5-12.

NOTE   Note the qualification specification which runs the filter only if the field (Status in our example) was modified in the current transaction and the modification was not made by the account which is used by the Remedy SPI server (RemSPI in our example). For this filter to be usable, you have to use different accounts than RemSPI for the interactive AR users.

The qualification expression for our example is ('Status' != 'DB.Status') AND ($USER$!= "RemSPI")

Figure 5-12   **The special filter for modifcations of the status field**

2. Add a run-process action that calls the OV SPI for Remedy client as
    shown in Figure 5-13.

NOTE   Note the keyword IF_MODIFIED and the ID of the modified field in the
                 command line of the Run Process.

The command line for our example is C:\temp\itoupdate.exe
$managed by$ ENTRY_ID=$Case ID$ SERVER=$SERVER$
SCHEMA="$SCHEMA$" OP=$OPERATION$ MODIFIED=7

Figure 5-13   **The special filter's Run Process**
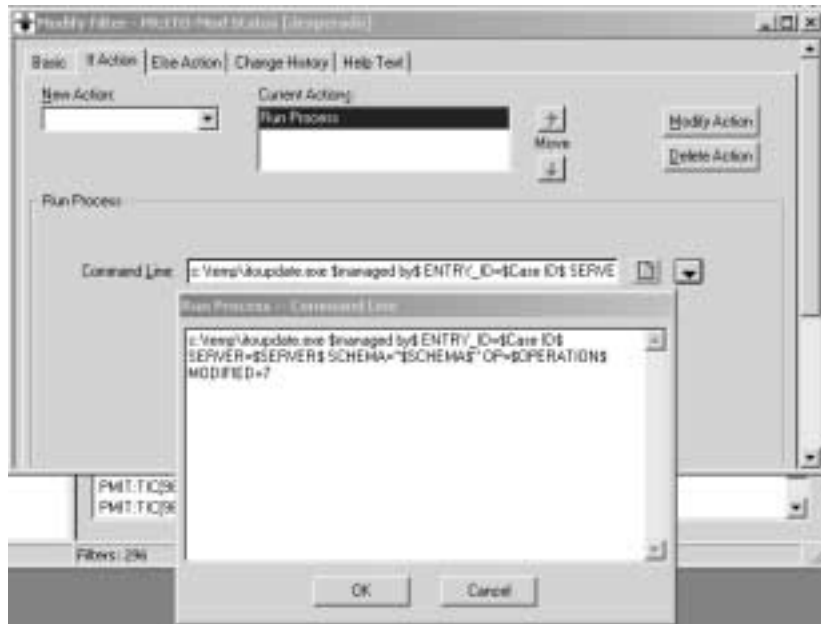
Configuring an Active Link

To configure an active link that enables users to start actions or escalate
messages in OVO/Unix:

1. Use the ARS to create the appropriate buttons or menu items in the
   ARS schema as shown in Figure 5-14.

Figure 5-14   **Buttons for Active Links**

2. Create a new active link that runs when someone uses the menu item

or button created in step 1.

Figure 5-15   A New Active Link

3. Add a run process that calls the OV SPI for Remedy client as
    illustrated in Figure 5-16. Note that the CMD parameter must have a
    value that matches the parameter of the OP_ACTION for this target.
    For more details, see "Defining Update Rules" on page 100.

Figure 5-16   **An Active Link's Run Process**
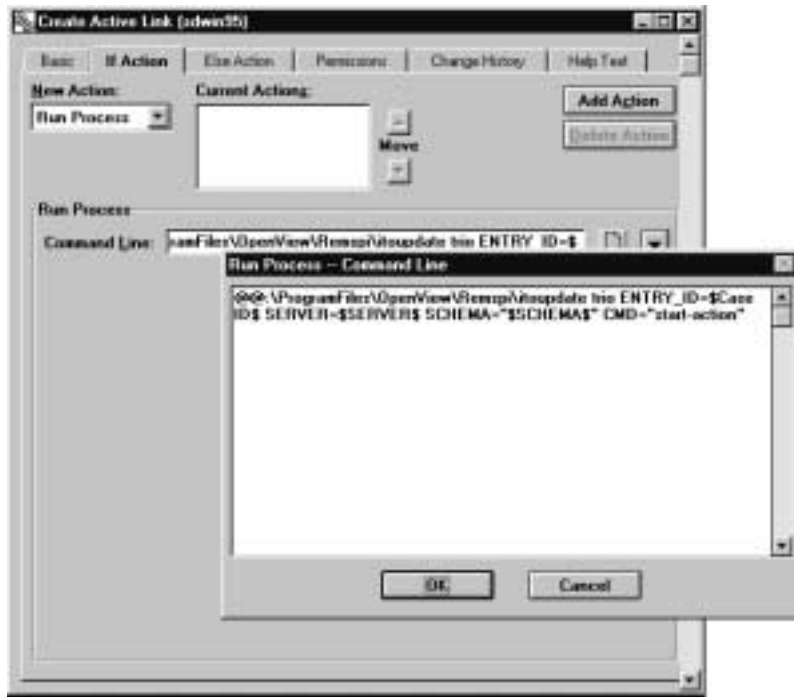
Customizing the OV SPI for Remedy for Multiple Servers

The OV SPI for Remedy enables you to use more than one AR server, more than one OVO/Unix server, or both. If any of these situations applies for your organization, you need to customize the OV SPI for Remedy appropriately. Figure 1-7 on page 33 shows an environment with multiple OVO/Unix and AR servers.

Using Multiple AR Servers

If you want the OV SPI for Remedy server to create and update action requests on more than one AR server, create appropriate targets in the rules file. Define different AR servers using the target's SERVER attribute. For more details on defining targets, see "Defining Targets" on page 99.

Example 5-2   **Target Definition with Multiple AR Servers**

```
TARGET "ProbSysUSA"
  SERVER "ars2.cnd.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "43B85572929960F64242C4D030C2CCABB65C236A"

TARGET "ProbSysEUR"
  SERVER "ars1.bbn.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "C338D5F22121EO76C200F6E6B4E94B2C679B2DA"
```

Example 5-2 shows two target definitions from a rules file. The first defines a target on an AR server in the USA. The second defines a target on an AR server in Europe.

To specify which AR server to use in which situation, define conditions with actions that use the appropriate target. For more details on defining conditions, see "Defining Conditions" on page 115.

Moving an AR ticket between AR servers

An AR ticket may be moved from one to an other AR server, this is

known as a DSO transfer and performed as an AR server function.

The actual move of the AR ticket is done by the AR server and not by the OV SPI for Remedy. The SPI is only notified after the AR server has completed the move in order to update the SPI databases.

In addition to the steps which have to be done in the AR server configuration (see the AR server documentation for details), the OV SPI for Remedy has to be informed about the ticket move. This has to be done in two steps:

- Notify the SPI server about the ticket move
- Instruct the SPI server to handle the move request

Notify the SPI server about the ticket move

The move notification is sent to the SPI server by executing the SPI client program itoupdate with a special parameter set. This execution of the SPI client can be configured as a filter or an active link bound to a button in an AR form.

Here is the syntax how the itoupdate program must be called in case an AR ticket has been moved between AR servers:

```
itoupdate \
  <OV SPI for Remedy server> \
   ENTRY_ID=<AR ticket ID on source server> \
    SERVER=<AR source server>  SCHEMA=<source form> \
  CMD=DSO-TRANSFER \
   ENTRY_ID=<AR ticket ID on destination server> \
    SERVER=<AR destination server> SCHEMA=<dest. form>
```

Instruct the SPI server to handle the move

The SPI server, receiving the move notification, will have to delete the association of the source ticket ID with the associated OVO messages and then have to add a new association between the new ticket ID and the same OVO messages. In addition, the information about the new AR server has to be saved for this association, too.

Remember that the SPI server saves the description of the condition which created an AR ticket. This "*creating*" condition is then later used to process manual attachments or changes to the OVO messages.

In case of a DSO transfer, the new ticket has not been created by the OV SPI for Remedy, but by the moving AR server. Thus, no such information about the *creating* condition is available, but this information is needed for later manual attachments or changes to associated OVO messages which is expected to result in a modification of the AR ticket.

This special "*creating*" condition has to be configured in the rules file in case of a DSO transfer.

```
DESCRIPTION "Catch All" DSO_TARGET
 CONDITION
  SET TARGET "ProbSys"
  ..
```

The condition which has to be used for later updates on the AR ticket is marked with the keyword DSO_TARGET.

If the SPI server receives a move notification, all conditions which have the target AR server in their SET TARGET list are checked for this keyword. The first condition which is found is saved as the "*creating*" condition.

NOTE   If no such condition is found, the old association between the source AR ticket and the associated OVO messages is removed, but no new association is established.

Using Multiple OVO/Unix Servers

If you want to create and update action requests based on OVO/Unix messages from more than one OVO/Unix server, you need further OV SPI for Remedy servers. Install one OV SPI for Remedy server on each machine where an OVO/Unix server runs.

If each of your OV SPI for Remedy servers has exclusive access to a set of AR schemas, you can customize the OV SPI for Remedy servers normally. Even if the schemas are on the same AR server, the OV SPI for Remedy instances function independently.

However, if more than one OV SPI for Remedy server accesses the same schema, you need to configure the schema, filter, and rules file appropriately. The OV SPI for Remedy client must know which OV SPI for Remedy server created an action request so that it contact the correct server with when someone changes the action request.

To enable this, you must complete the following steps for each schema:

1. Create a field in the schema to store the name of the OV SPI for Remedy server. This can be a hidden field if the AR system users do not need to see it.

2. Amend the filter that calls the OV SPI for Remedy client. Reference the new field as the first parameter in the OV SPI for Remedy client command.

   Figure 5-17 shows an example filter's run process where field SPI Server contains the name of the OV SPI for Remedy server that created the action request.

   The exact path and name for the OV SPI for Remedy client program depends on the platform and your setup. For more details, see "Installing the OV SPI for Remedy" on page 44.

Figure 5-17   Filter Run-Process for Multiple OVO/Unix Servers
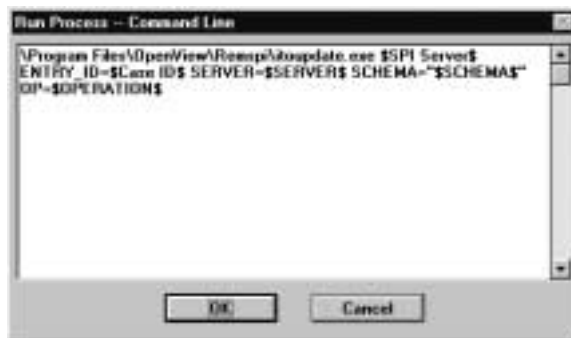
3. Amend the rules file for *both* OV SPI for Remedy servers. Each rules file must contain an action that puts the name of the OV SPI for Remedy server into the appropriate field using the action variable HOST_NAME.

   The action must use the FIRST keyword to ensure that the OV SPI for Remedy enters only one value. For more details about defining actions, see "Defining Actions" on page 121.

   Example 5-3 shows a default action for a target. The action puts the name of the OV SPI for Remedy server in field 1000000006.

Example 5-3   Default Action for Multiple OVO/Unix Servers

```
TARGET "ProbSysEUR"
```

```
SERVER "ars1.bbn.hp.com"
SCHEMA "Problem Report"
USER "spi_user"
PASSWORD "C338D5F22121EO76C200F6E6B4E94B2C679B2DA"

FIRST FIELD 1000000006 "$HOST_NAME$"
```

Using the OV SPI for Remedy on cluster nodes

If the OV SPI for Remedy is used on cluster nodes, several configuration issues have to be considered.

NOTE   The following rules only describe the SPI specific requirements which have to be satisfied to use the OV SPI for Remedy on cluster nodes. For detailed explanations on how to set up the OVO management server and/or the OVO managed node on cluster nodes, see the appropriate OVO documentation.

AR Server installed on a cluster node

If the Remedy AR server is running on a cluster node, we will refer to it as the *AR package*. In this situation, the following areas are impacted:

- Monitoring

  — AR package is made active

    The following commands have to be inserted in the package switching script:

    — opctemplate -e "RemSPI_ARS_server_UX"

    — opctemplate -e "RemSPI ARS Server errors (UX)"

  — AR package is made inactive

    The following commands have to be inserted in the package switching script:

    — opctemplate -d "RemSPI_ARS_server_UX"

    — opctemplate -d "RemSPI ARS Server errors (UX)"

- Communication OVO server to AR server

  In the rules file, use the *package name/IP* as the name of the TARGET.

- Communication AR server to OVO server

The filters and active links on the AR server starting the SPI client `itoupdate` have to pass the *AR package name/IP* for the sender ID as a parameter.

OVO Server installed on a cluster node

If the OVO management server is running on a cluster node, we will refer to it as the *OVO package*. In this situation, the following areas are impacted:

- Monitoring

  — AR package is made active

  The following commands have to be inserted in the package switching script:

  — `opctemplate -e "RemSPI_QSize"`

  — `opctemplate -e "RemSPI_Server"`

  — AR package is made inactive

  The following commands have to be inserted in the package switching script:

  — `opctemplate -d "RemSPI_QSize"`

  — `opctemplate -d "RemSPI_Server"`

- Communication AR server to OVO server

  The filters and active links on the AR server starting the SPI client `itoupdate` have to pass the *OVO package name/IP* for the destination address as a parameter.

Using External Actions and Data

The first time the OV SPI for Remedy submits a OVO/Unix message to the AR system, you can use external data and start external actions. However. we strongly recommend that you use this feature if, *and only if*, you are an experienced shell programmer. This section covers the following areas:

- Starting External Actions

- The remspiupd Command

- Using External Value Files

- Formatting External Value Files

- Referencing External Values

Starting External Actions

To submit a OVO/Unix message to the OV SPI for Remedy, the scripts `remspisub_auto.sh` (Trouble Ticket interface) and `remspisub_man.sh` (OVO/Unix application) pass the message ID to the `remspiupd` command, which forwards the information to the OV SPI for Remedy server process. The two scripts, both of which are located in `/opt/OV/bin/remspi/`, have a special `User Code` section, to which you can add commands. For example, you could include commands to start any external actions that are necessary. Similarly, the commands could trigger actions to update or retrieve data in an external database. For example, you could use the name of a node and the object type to find out who is responsible for a printer that is giving problems. The information you retrieve could even include the responsible person's telephone number.

The user code section could also include actions that generate external value files. External value files contain additional information that the OVO/Unix message does not provide. The OV SPI for Remedy can add this extra information when it creates a new action request.

The remspiupd Command

The `remspiupd` command handles the internal communication between the HP OV SPI for Remedy server and OVO/Unix. The command can be

used both to submit and update OVO/Unix messages and is called by the
`remspisub_*.sh` scripts. For more information on the `remspisub_*.sh`
scripts, see "Data Components" on page 33.

The `remspiupd` command accepts the following command line
arguments:

```
remspiupd  [-d][-t][-q][-a] \
           [-u <OVO user submitting the message>] \
           [-i <instruction-text>] \
           [-f <external value file>] \
           <ITO message ID> [ … ]
```

where at least one OVO/Unix message ID must be specified, and the:

`-a(ttach)`	enables one message to be *attached* to another message. Note that at least *two* OVO/Unix message IDs must be specified.
`-d(ebug)`	option writes *debugging* information to `stdout`
`-t(race)`	option writes *trace* information to `/var/opt/OV/share/tmp/remspiupd.trace`
`-i(nstr)`	option allows you to specify an instruction text, which may be referred to with the `HLP_TEXT` action keyword during processing
`-f(ile)`	option allows you to specify an external value `file` which may contain arbitrary values that are used during processing
`-q(uiet)`	option enables *quiet* mode. Normally the `remspiupd` program waits for a reply from the HP OV SPI for Remedy server and prints this to `stdout`. This is intended for submission from the OVO/Unix GUI to immediately display the result of the submission. The default submission scripts use the `-q` option for automatic submission by the OVO/Unix server.
`-u(user)`	option allows to specify the OVO user submitting the message

If you are using the `remspiupd` command to submit a message, *no*
OVO/Unix messages must already be assigned to the AR ticket. If you
are using the `remspiupd` command to *update* messages, the whole set of
message IDs which you specify *must* meet the following conditions:

- at least *one* OVO/Unix message must already have been submitted

- the already submitted OVO/Unix messages must be associated with only *one* AR

- there must be at least *one* OVO/Unix message which has *not* yet been assigned

These messages will be attached to the AR associated with the other message(s).

If any one of these conditions is *not* met, the remspiupd command returns an error and the message submission fails. The command remspiupd exits with the value 0 if the submission succeeds and a non-0 value if the submission fails. Note that the remspiupd command writes error information to the file remspiupd.log. For more information on the contents of the remspiupd.log file and where to find it, see "Error Logs and Tracing" on page 171.

If the option -q is not specified and new AR tickets have been created for the OVO message(s), the following information is printed to *stdout* for each AR ticket - on separate lines:

<AR server>:<AR Form>:<AR Id>.

Using External Value Files

The HP OV SPI for Remedy allows you to use external values as a source of information in a similar way to the attributes and fields of a OVO/Unix messages. These external values are stored in an external file, an example of which is located in:

/var/opt/OV/share/tmp/remspi/extval

You can use external value files to add information to an action request that is not available in the original OVO/Unix message. An external value file can either:

- be static and rarely change

- be generated dynamically using external actions

NOTE   You need to specify the name of the external value file. In addition, you can use variables in the shell script to access one of a number of files, for example; depending on the node. How you go about doing this depends to

a very large extent on what parameters are available in the shell script.

The name of the external file is stored in the User Code section of the scripts remspisub_auto.sh and remspisub_man.sh, which submit the OVO/Unix message to the HP OV SPI for Remedy server (and thus, to the AR server). If you need to configure dynamic access to the external values you define, these scripts should calculate the required values, write them to a text file, and pass the location of this text file to the HP OV SPI for Remedy server by setting the script variable *EXT_VAL_FILE* in the User Code section.

Formatting External Value Files

All external value files must consist of key and value pairs with the following format rules:

- The file contains plain text.

- Each key / value pair appears on a separate line.

- The key consists of any printable characters, but the size of the key is limited to 2048 characters.

- At least one space or tab separates each key and value.

- The value is a string with any characters (including spaces), the size of this string is limited to 4096 characters.

Example 5-4 shows a simple external value file that contains extra details about a printer.

Example 5-4   **External Value File**

```
# Last updated 11/12/1999

LOCATION       Building 4 - Level 1
CONTACT_NAME   Wilson, J
CONTACT_EXT    2345
```

NOTE   An external value file can contain blank lines and comments. The first character of a comment line must be #.

In the rules file, the length of a token (a string is a token, too) is limited to 2048 characters. Hence, if the value of an external key is used in a

condition, its length must not exceed 2048 characters in the `rules` file (if it's longer, consider using a regular expression).

Referencing External Values

To add external values to an action request, you must reference them in actions within the rules file by using the following variable-keyword combination:

EXTERNAL <*external key*>

You use this like any other variable in the rules file, except that it consists of two keywords. For more details on actions, see "Defining Actions" on page 121.

Example 5-5 shows an action that puts external data from Example 5-4 into a text field. Figure 5-18 shows the result in the action request.

Example 5-5   **References to External Values**

```
    FIELD 1000000008 "Contact Details \\n
* Location:        $EXTERNAL LOCATION$
* Contact Name:    $EXTERNAL CONTACT_NAME$
* Extension:       $EXTERNAL CONTACT_EXT$"
```

Figure 5-18   **External Values in an Action Request**

NOTE   The OV SPI for Remedy only evaluates external values when it creates the action request. You cannot use external values to update existing action requests.

6 Troubleshooting the HP OV SPI for Remedy

This section describes how to go about troubleshooting the HP OpenView Smart Plug-In for Remedy Action Request System integration. In this section you will find information concerning:

- Error Logs and Tracing
- Troubleshooting the HP OV SPI for Remedy Installation
- Troubleshooting HP OV SPI for Remedy Rules
- Troubleshooting HP OV SPI for Remedy Usage

Error Logs and Tracing

This section describes the log files to which the HP OV SPI for Remedy server writes information concerning errors and tracing. Table 6-1 on page 171 lists the log files that you can examine as a first step in the event of any serious problem and indicates where the files can be found.

Table 6-1 **Logfile Locations**

File Name	HP OV SPI for Remedy System	Location
itoupdate.log	Client	/var/opt/OV/log/remspi \usr\OV\log\remspi If not installed on an OVO managed node, same directory as itoupdate command
remspi.log	Server	/var/opt/OV/log/remspi
remspiupd.trace		/var/opt/OV/share/tmp/remspi
remspi_submit.log		/var/opt/OV/log/remspi
trace		/var/opt/OV/share/tmp/remspi

The itoupdate command on the HP OV SPI for Remedy client writes error messages to the itoupdate.log. The itoupdate.log itself is written to the directory in which the itoupdate command is installed if the standard log directory is not found, i.e. if the client system is not an OVO managed node. The path is determined by the execution of the AR filter itself. For example, if the AR filter executes /abc/itoupdate, then the log file will be written to /abc/itoupdate.log.

The remspi.log contains error messages from the HP OV SPI for Remedy server and information relating to the self-monitoring of the HP OV SPI for Remedy server. For more information, see "Troubleshooting HP OV SPI for Remedy Usage" on page 178.

The remspiupd.trace contains trace information relating to attempts

by the HP OV SPI for Remedy to submit OVO/Unix messages to the ARS and attach modifications to existing OVO/Unix messages. For more information, see "The remspiupd Command" on page 164. Note that this file is not written by default.

The remspi_submit.log contains information relating to the manual or automatic submission of OVO/Unix messages to the ARS. For more information, see "Troubleshooting HP OV SPI for Remedy Usage" on page 178.

The trace file contains information relating to the internal control flow of the SPI programs, especially the internal processing in the SPI server.

You can control the amount of information written to the trace in the HP OV SPI for Remedy configuration file remspi.cfg.

The keywords used for tracing are listed in Table 6-2 on page 172.

Table 6-2   **Keywords for Tracing**

Keyword	Parameter
SPI_TRC_LEVEL	An integer value ranging from 1to 9.
	Level 1 emits less messages, level 9 the emits the most information.
	For support purposes, level 5 is the mostly sufficient.
	Level 6 adds detailed information about the usage of the Remedy API calls, i.e. all input and output parameters are listed together with the result of the API call.

To turn tracing on, you have to follow these steps:

- Add/uncomment the line SPI_TRC_LEVEL 5 in the remspi.cfg file (use a trace level according to your needs)

- Restart the SPI server using the command sequence
  ovstop RemedySPI; ovstart RemedySPI

As an alternative, you may send a signal to the running SPI server after having updated the remspi.cfg file. This will force the SPI server to re-read it's configuration file remspi.cfg and to switch the trace mode accordingly:

- HP-UX: kill -USR2 *<pid of remspisrv process>*

- Solaris: `kill -PROF <pid of remspisrv process>`

To turn tracing off, you have to follow these steps:

- Delete/comment the line `SPI_TRC_LEVEL` *n* in the `remspi.cfg` file

- Restart the SPI server using the command sequence
  `ovstop RemedySPI; ovstart RemedySPI`

As an alternative, you may send a signal to the running SPI server after having updated the `remspi.cfg` file. This will force the SPI server to re-read it's configuration file `remspi.cfg` and to switch the trace mode accordingly.

NOTE   Essential information is written to the trace file if the trace mode is turned on before the SPI server is started. Thus, this should be the preferred way to produce a trace file.

However, if it's not possible to have the trace turned on for a very long time, then start the SPI server with trace enabled (to catch the startup information), then turn off the trace mode for the running SPI server and enable the trace again shortly before the situation you want to inspect occurs.

If you want to see what is happening during the parsing of the `rules` file, you can start the HP OV SPI for Remedy server process manually using the following command. Enter:

/opt/OV/bin/remspi/remspisrv -debug

This keeps the `remspisrv` process in the foreground and prints trace information to `stderr` during the parsing phase of the `rules` file.

Troubleshooting the HP OV SPI for Remedy Installation

To verify that the installation of the HP OpenView Smart Plug-In for Remedy Action Request System integration completed successfully on the OVO/Unix management server, carry out the steps in the following list:

1. On the command line, enter: `swlist`

   Look for the following entries:

   `SPI-Remedy A.02.50`

   `SPI-RemedyDoc A.02.50`

2. In the OVO/Unix GUI, check that the following elements have been added:

Node Groups	`Remedy SPI Servers`
	`AR Servers NT`
	`AR Servers UX`
Message Groups	`Remedy-SPI`
	`AR System`
Application Groups	`Remedy SPI Admin`
	`Remedy SPI`
Users	`remspi-adm`
	`remspi-op`
Message-Source Templates	`OV SPI for Remedy - AR Server NT`
	`OV SPI for Remedy - AR Server UX`
	`OV SPI for Remedy - ITO`

3. If either of these steps fails to produce the desired results:

   a. Check the following log files for more specific information relating to problems concerning the installation of the HP OV SPI for

Remedy:

- `/var/adm/sw/swagent.log`

- `/var/adm/sw/swinstall.log`

b. Cross check your installation with the information provided in Table 2-8, "File Locations on the OVO/Unix Management Server," on page 51.

NOTE After you install the OV SPI for Remedy client program on an AR server, you do not normally need to configure it. However, if the default TCP port 13603 is not available for the OV SPI for Remedy server, you will need to configure an alternative in the `/etc/services` file on the OV SPI for Remedy server machine - the OVO/Unix management server.

To modify port a number:

1. As user `root` on the OVO/Unix management server, open the `/etc/services` file in a text editor.

2. Find the section that begins with the comment:

   `# HP OpenView OVO/Unix Remedy SPI entries`

3. Edit the TCP port number as necessary. For example, if port 42 should be used, enter:

   `remspi 42/tcp`

   Note that the same change must be implemented on the AR server system, too.

Troubleshooting HP OV SPI for Remedy Rules

The rules file determines how the HP OV SPI for Remedy should react to messages from OVO/Unix that are intended for ARS, and action requests from the AR system, which are intended for OVO/Unix. For more information about the syntax and structure of the rules file, see "Customizing the Rules File" on page 95.

NOTE
The HP OV SPI for Remedy server will not start if the rules file contains any errors. To check for errors, use the remspiconfig command, which is located in the following directory /opt/OV/bin/remspi.

The remspiconfig command parses the rules file and writes the output to a terminal window. The Validating... section gives details of any errors that have been encountered. Example 6-1 shows the messages that appear when the rules file references a non-existent schema field.

Example 6-1   **remspiconfig Output**

```
Remedy SPI configuration utility
================================

Loading rules file '/etc/opt/OV/share/conf/remspi/rules' ...
Reading rules file '/etc/opt/OV/share/conf/remspi/rules' ...

Validating ...
01/30/2003 10:41:40 [ERROR] (remspiconfig-26082): Can't
initialize rules table from file
'/etc/opt/OV/share/conf/remspi/rules'. (SPI215-50)
Can't validate target 'ProbSys'. (SPI215-66)
Can't validate AR System field '22' of target 'ProbSys'.
(SPI215-62)
ARS error[0]: 314: 'Field does not exist on current form'.
(SPI215-61)

Connecting ITO as 'opc_adm' ...
Connection succeeded.

All done.
```

The lines beginning with Can't validate... enable you to locate the error in the rules file. In Example 6-1, the error is a reference to field 22

in the `ProbSys` target.

Lines that begin with the words `ARS error` describe error messages that the AR system returns to the HP OV SPI for Remedy. The text that appears gives a brief explanation of the error. The number after the term ARS Error and preceding the error description is the AR System error number (in this example; `314`). For more information about the error, use the error number, for example `314`, to refer to the AR System documentation.

NOTE   If the `remspiconfig` command is started while the SPI server is running, an error message is printed indicating the fact that some files could not be locked.

This is because the SPI server as well as the `remspiconfig` command try to lock the SPI databases.

Troubleshooting HP OV SPI for Remedy Usage

This section describes some of the problems that can occur when using the HP OpenView Smart Plug-In for Remedy Action Request System integration. Some of the problems are indicated by the arrival of a message in the OVO/Unix message browser. The section covers the following areas:

- HP OV SPI for Remedy Monitors

- Name Resolution

- Responsibility of the REMSPI_ITO_USER

- Removing buffered AR requests

- Setting the language of the SPI server

- Duplicated annotations

HP OV SPI for Remedy Monitors

Symptom   If you acknowledge (or take ownership of) a large number of OVO/Unix messages that are linked to ARS, the overall performance of the HP OV SPI for Remedy can be significantly affected.

Explanation   The HP OV SPI for Remedy provides a number of self-monitoring processes, which can trigger an alert as soon as an issue arises and, as a result, allow timely intervention in the case of problems. One of the monitors watches the size of the OVO/Unix request queue, RemSPI_QSize, on the HP OV SPI for Remedy server. RemSPI_QSize holds HP OV SPI for Remedy-related OVO/Unix messages with a severity level of either warning or critical. If the number of messages in RemSPI_QSize exceeds 1000, the HP OV SPI for Remedy generates a message. Note that self-monitoring error messages are written to the log file; remspi.log.

Name Resolution

Symptom   Problems appear to occur during the resolution of IP addresses on AR servers with multiple host names (multi-homed hosts).

Explanation   You can use short host names, long host names, or the IP address of the AR system when defining AR server names within the HP OV SPI for Remedy. However, the HP OV SPI for Remedy uses a specific order when resolving these names, namely:

1. short hostname

2. long hostname

3. IP address

For more information on defining an AR server in HP OV SPI for Remedy target definitions, see "Defining Targets" on page 99.

Responsibility of the REMSPI_ITO_USER

Symptom   Error occurs when message are submitted (manually or automatically). There may be multiple symptoms depending on the construction of the rules file. Among them are: error from the AR server about missing, but required fields or invalid field values, the constructed AR ticket is missing field values, etc.

Explanation   The OVO user which is configured as the REMSPI_ITO_USER in the remspi.cfg file has to be responsible for all the OVO messages which are to be submitted to the AR system. Failure to do so will result in error messages written to the log file and shown as the application output (for manually submitted messages).

Removing buffered AR requests

Symptom   If you want to clear the list of buffered AR requests (if `OPTION BUFFER_TT` is specified), then you have to empty the queue directories and restart the SPI server.

Explanation   The buffered AR requests are stored in directories
`/var/opt/OV/share/tmp/remspi/queue.mce` and
`/var/opt/OV/share/tmp/remspi/queue.sub`. To
delete the buffered requests, follow these steps:

- ovstop RemedySPI

- rm -f /var/opt/OV/share/tmp/remspi/queue.mce/*

- rm -f /var/opt/OV/share/tmp/remspi/queue.sub/*

- ovstart RemedySPI

Setting the language of the SPI server

Symptom   If you experience problems in creating new tickets on
an AR server and the server rejects the requests with
complaints about illegal user names or passwords, it
may be that the SPI server is running in a language
environment not supported by the AR server. This may
be checked with trace level 6 and inspecting the return
parameters of the AR API calls.

Explanation   To start the SPI server in a specific language
environment, you may configure the language in the
remspi.cfg file.

To start the SPI server in a "C" language environment,
add the following line to the remspi.cfg file and restart
the SPI server:

`REMSPI_LANG C`

Duplicated annotations

Symptom   If you see multiple, duplicated annotations to OVO
messages or duplicated entries in AR tickets, you may
decide to reconfigure your rules file or to establish a
new AR user performing the AR ticket updates.

Explanation   If the rules file is configured in such a way that
changes to AR tickets are added as annotations to the
OVO messages and changes to the OVO messages are
treated as an update request for the AR tickets, then it
may happen that some changes are duplicated on one
or both sides. This is because the OV SPI for Remedy is

not able to determine whether the AR ticket was updated manually or by the SPI itself.

To help the SPI to distinguish between manual and automatic (SPI driven) updates, you can use a special AR user for all SPI transactions. This AR account is then not used for interactive users. See "Configuring a special IF_MODIFIED ARS Filter Manually" on page 152 for details on how to configure.

Troubleshooting the HP OV SPI for Remedy Internal Database

The HP OV SPI for Remedy stores the association between OVO/Unix messages and AR tickets in an own database. This database can be maintained and verified using the utility *remspidbutil*.

Usage:

```
remspidbutil -help |
             -verify [ -target target ... ] |
             -delete [ -target target ... ] [ -ar ] ID ...
```

where

-help   shows the usage information for the tool

-verify   is used to verify the internal database. If a target (see the rules file) is specified, then only the database entries for this target are checked, else all targets defined in the rules file are checked.

-delete   is used to delete entries from the internal database. If the option -ar is specified, then the AR ticket ID's are expected as ID ... arguments, else OVO/Unix message id's are expected. If explicit targets are given, the entries are deleted only when occurring in these associations.

7 De-installing the HP OV SPI for Remedy

This section describes how to de-install the HP OpenView Smart Plug-In for Remedy Action Request System integration software from the systems on which you have installed it. The section covers the following areas:

- Removing AR Filters
- Removing the HP OV SPI for Remedy from the OVO/Unix GUI
- Removing the HP OV SPI for Remedy from the OVO/Unix Management Server
- Removing HP OV SPI for Remedy Software from the AR Server

Removing AR Filters

To remove or unregister AR System filter, use the `remspifilter` command with the `-d(elete)` option. For example:

remspifilter -delete *<filter name>*

For more information about how to use the `remspifilter` command and which options the command accepts, see "Configuring the ARS Filter Automatically" on page 150.

NOTE   You can use the ARS administrator's GUI to see a complete list of the filters.

NOTE   If you have defined special IF_MODIFIED filters manually, you have to remove them manually using the ARS administrator's GUI. The command `remspifilter` is not able to delete these filters from the AR server.

Removing the HP OV SPI for Remedy from the OVO/Unix GUI

The HP OV SPI for Remedy integration with the OVO/Unix GUI has to be removed manually. You will have to remove the following components from the OVO/Unix GUI:

- Managed Nodes Groups

  Note that you should not remove the AR servers from the Node Bank window until you have removed the OVO/Unix agent software including the HP OV SPI for Remedy templates. For more information, see "Removing HP OV SPI for Remedy Software from the AR Server" on page 188.

- Message Groups

- Applications and Application Groups

- Templates and Template Groups

- Users

For more information on how to remove elements from the OVO/Unix GUI, see the *HP OpenView Operations Administrator's Reference*.

De-installing the HP OV SPI for Remedy
**Removing the HP OV SPI for Remedy from the OVO/Unix Management
Server**

Removing the HP OV SPI for Remedy from the OVO/Unix Management Server

To remove the HP OpenView Smart Plug-In for Remedy Action Request System integration components from the OVO/Unix management server and complete the general clean up process:

1. On the OVO/Unix management server, use the swremove command. Enter:

   swremove SPI-Remedy

2. De-assign the HP OV SPI for Remedy templates from the OVO/Unix management server and the AR server systems:

   a. In the OVO/Unix Node Bank window or the appropriate HP OV SPI for Remedy node-group window, select the OVO/Unix management server and the HP OV SPI for Remedy systems.

   b. Distribute the *empty* template-assignment list to the selected nodes using the following menu sequence.

      Actions:Agents-> Install/Update SW & Config...

3. Cleanup the GUI by removing the HP OV SPI for Remedy-specific elements. For more information, see "Removing the HP OV SPI for Remedy from the OVO/Unix GUI" on page 186.

Removing HP OV SPI for Remedy Software from the AR Server

After you have removed from the OVO/Unix GUI those elements added by the HP OV SPI for Remedy and removed the HP OV SPI for Remedy software from the OVO/Unix management server, you need to remove the OVO/Unix agent software, including the HP OV SPI for Remedy templates, from the AR server.

To remove OVO/Unix agent software, including the HP OV SPI for Remedy templates from the AR server, open the Node Bank or Node Group Bank window, select the managed node or node group from which you want to remove the OVO/Unix agent software, and use the menu item:

```
Actions:Agents -> De-install
```

To remove *only* the HP OV SPI for Remedy templates from the AR server, re-distribute the OVO/Unix agent software *without* the HP OV SPI for Remedy templates to the AR servers which you want to clean up. OVO/Unix recognizes that there is no record of the HP OV SPI for Remedy on the OVO/Unix management nodes you have selected and overwrites (with nothing) the existing HP OV SPI for Remedy installation—effectively removing the HP OV SPI for Remedy templates from the AR server. For more information about how to distribute templates in OVO/Unix, see "Distributing the OV SPI for Remedy Templates" on page 50.

After you have removed the HP OV SPI for Remedy installation on the AR servers, you can remove the AR servers from the Node Bank window of the OVO/Unix GUI.

Index

Index

Index

Index

Index

Index