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

Administrator's Reference

Version B.01.10

MS Windows®

This PDF file contains the same information found in the online help. Some interactive pages are not included.

Manufacturing Part Number: None

March 2005

Legal Notices

Warranty.

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

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

Restricted Rights Legend.

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

Hewlett-Packard Company
United States of America

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

Copyright Notices.

Trademark Notices.

Intel®, Itanium®, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Windows® 2000 is a U.S. registered trademark of Microsoft Corporation.

Contents

Contents

Contents

Support

Please visit the HP OpenView web site at:

http://openview.hp.com/

This web site provides contact information and details about the products, services, and support that HP OpenView offers. You can also go directly to the support web site at:

http://support.openview.hp.com/

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. To find more information about access levels, go to the following URL:

http://support.openview.hp.com/access_level.jsp

To register for an HP Passport ID, go to the following URL:

https://passport.hp.com/hpp2/newuser.do

1 Smart Plug-in for Remedy ARS

This section describes what you can find in the online Help for the SPI for Remedy ARS.

In this Section

The on-line Help for the HP OpenView Smart Plug-In for Remedy Action Request System integration provides all the information you need to install and use the SPI for Remedy ARS to manage and monitor your Remedy ARS environment from a central location. In the on-line Help, you will find information about the following topics:

- "Introducing the SPI for Remedy ARS"

  This section introduces the SPI for Remedy ARS and explains some of the functionality.

- "Installing the SPI for Remedy ARS"

  This section describes how to install and, if necessary, upgrade the SPI for Remedy ARS.

- "Getting Started"

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

- "Using the SPI for Remedy ARS"

  This section describes the SPI for Remedy ARS interface that you will use in the performance of your every-day tasks.

- "Customizing the SPI for Remedy ARS"

  This section describes how to customize the HP OpenView Smart Plug-in for Remedy ARS to suit the demands of your environment.

- "Troubleshooting the SPI for Remedy ARS"

  This section describes how to go about troubleshooting the HP OpenView Smart Plug-In for Remedy Action Request System integration.

- "De-installing the SPI for Remedy ARS"

  This section describes how to de-install the SPI for Remedy ARS software from the systems on which you have installed it.

2 Introducing the SPI for Remedy ARS

This section introduces the **HP OpenView Smart Plug-In for Remedy Action Request System integration** software.

In this Section

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 SPI for Remedy ARS

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

Figure 2-1 Integration between OVO and an AR System

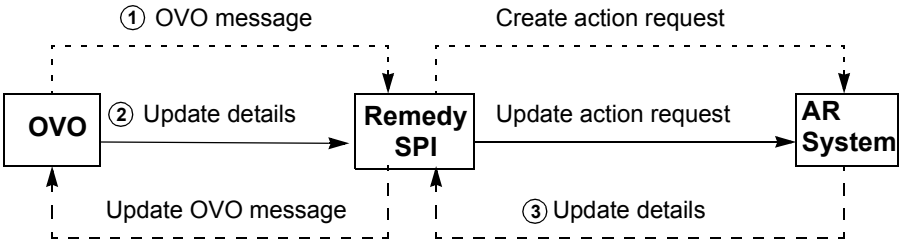


Figure 2-1 gives an overview of how the SPI for Remedy ARS integrates OVO and an AR System. It shows that:

1. The SPI for Remedy ARS accepts messages from OVO, then forwards the OVO message to the AR System. This creates a new action request which corresponds to the OVO message.
2. If an OVO message is modified, the SPI for Remedy ARS updates the action request.
3. If an action request is modified, the SPI for Remedy ARS updates the OVO message details.

For example, if OVO receives a message about a printer failure, the SPI for Remedy ARS 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 SPI for Remedy ARS changes the OVO message as appropriate. After the engineer repairs the printer, they update the status of the action request. The SPI for Remedy ARS then changes the status of the message in OVO.

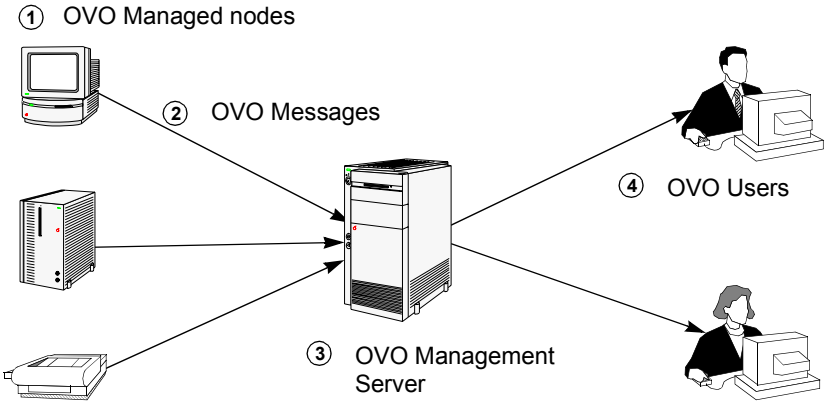
The rest of this chapter describes the SPI for Remedy ARS's functionality and architecture in more detail.

Understanding OVO

This section gives a brief overview of the OVO features that are important for the SPI for Remedy ARS. For more detailed information about OVO, see the *HP OpenView Operations Concepts Guide*. This section contains information on the following topics:

- "Messages"
- "Message Management"
- "Actions"

Figure 2-2 Overview of OVO



Messages

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

1. OVO can monitor and control remote systems, known as managed nodes.
2. If a particular status or event occurs on a managed node, the OVO 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 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 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.
- *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 2-2 shows how messages enable OVO 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 SPI for Remedy ARS. 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 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.

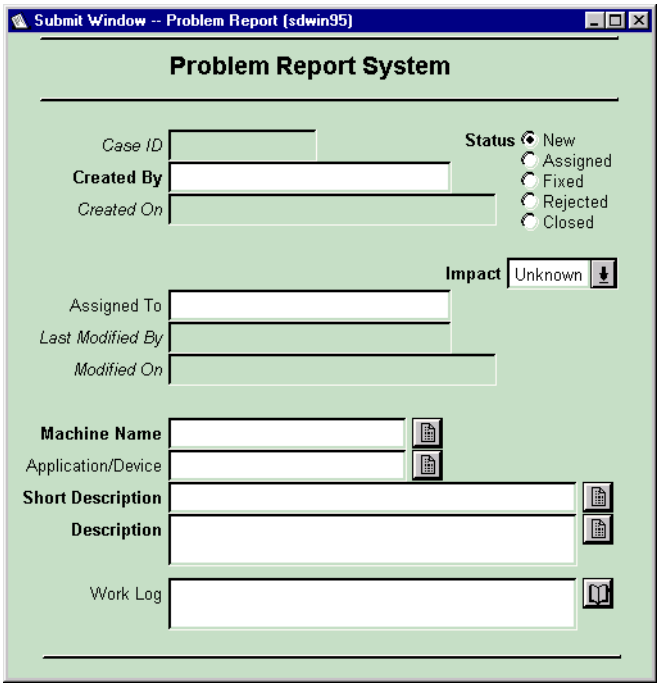
The AR Form

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. This section also explains the following related concepts:

- "AR Filters" on page 21
- "AR Active Links" on page 21

Figure 2-3 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.

Figure 2-3 A Simple AR Schema



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 SPI for Remedy ARS 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 SPI for Remedy ARS uses the AR filter mechanism to trigger communication between the ARS and HP OpenView OpenView Operations for Windows. The SPI for Remedy ARS 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 can trigger automatic updates in the OVO database via the SPI for Remedy ARS. The SPI for Remedy ARS 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 147.

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 SPI for Remedy ARS uses this feature to provide buttons in AR forms which allow the user to launch operator-initiated actions in OVO from within ARS or even escalate OVO 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 147.

The Functionality of the SPI for Remedy ARS

The SPI for Remedy ARS enables you to integrate OVO's monitoring and control capabilities with an AR System. When OVO detects problems in the environment, the SPI for Remedy ARS can immediately and automatically create an action request even before the first human notices the situation. In addition, the SPI for Remedy ARS maintains consistency between action requests and OVO messages until the problem is solved. This section also provides information about the following topics:

- "Key Features" on page 23
- "Example Scenarios" on page 24

Key Features

The SPI for Remedy ARS can:

- adapt to fit an organization's individual needs
- create new action requests by automatically sending a OVO message to an AR System
- create new action requests when an OVO user decides to send one or more OVO messages to an AR System (only OVO for UNIX)
- use external data and actions to add supplementary details to an action request
- update action requests when the corresponding OVO message changes
- update action requests by adding further OVO messages
- update OVO messages from the AR System. This includes:
  - changing ownership of the OVO message
  - adding annotations to the OVO message
  - escalating a OVO message
  - acknowledging a OVO message to reflect the status of the action request
  - starting an action in OVO

The SPI for Remedy ARS also provides:

- complete integration with the OVO GUI
- the capacity to monitor AR system availability with OVO
- the capacity to monitor SPI for Remedy ARS availability with OVO

Example Scenarios

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

Figure 2-4 An Action Request that Provides Information

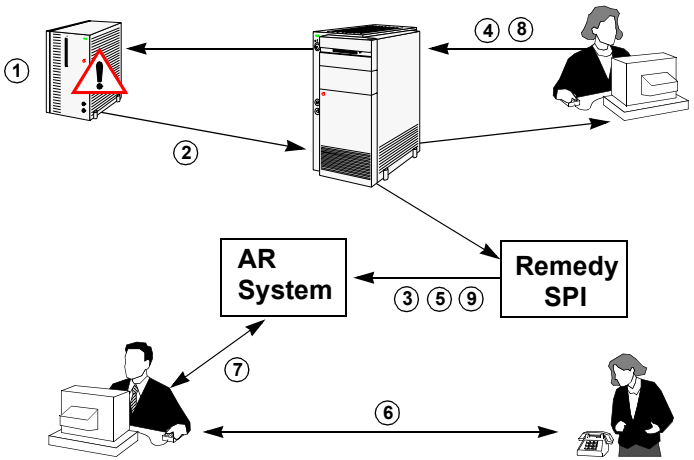


Figure 2-4 shows an example of how the SPI for Remedy ARS can create and update an action request to provide information.

1. A backup procedure fails because the disk is full.
2. The OVO agent on the managed node sends a message to the OVO management server.
3. The OVO management server automatically tells the SPI for Remedy ARS to create a new action request via the OVO Trouble-Ticket interface.

4. The OVO 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 message.
5. The SPI for Remedy ARS 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 user solves the problem of the full disk and acknowledges the OVO message.
9. The SPI for Remedy ARS automatically updates the information associated with the action request and the action request's status.

Figure 2-5 An Action Request that Needs Attention

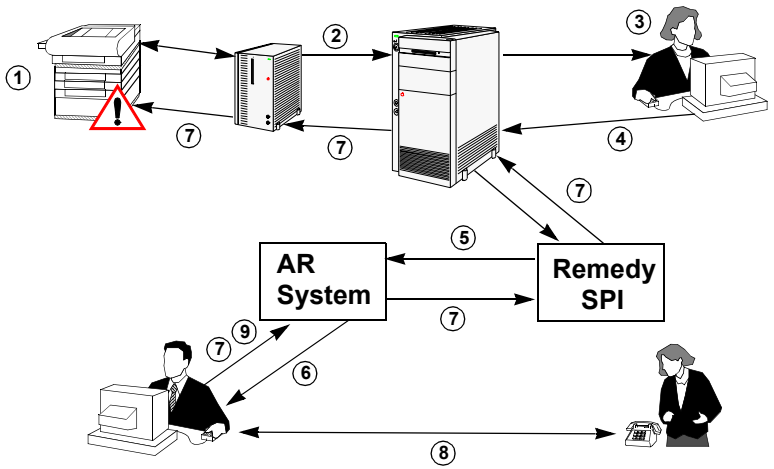


Figure 2-5 shows an example of how the SPI for Remedy ARS can create an action request that needs attention, then update the OVO message when the action request changes. Note that this scenario is available on OVO Unix management servers only.

1. A large printer in the print room fails because of a paper jam.
2. The OVO agent on the managed node sends a message to the OVO management server at a different site.

3. The OVO operator sees the message and diagnoses the problem. The OVO operator adds an annotation to the OVO message explaining that someone needs to remove the jammed paper and then reset the printer.
4. Since the OVO operator cannot remove the jammed paper from a printer in a remote site, he decides to forward the OVO message to the AR System.
5. The SPI for Remedy ARS receives the OVO 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 operator `Remedy SPI User` automatically takes ownership of the OVO message linked to the Action Request. Ownership of an OVO message indicates to other OVO operators that the problem is being solved.
7. The help desk staff start an action in OVO 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 SPI for Remedy ARS automatically acknowledges the corresponding OVO message.

The SPI for Remedy ARS Architecture

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

- "Process and Network Components"
- "Data Components"
- "SPI for Remedy ARS Components Integrated with OVO"

Process and Network Components

To use the SPI for Remedy ARS, the components shown in Figure 2-6 at the very least must be present, namely:

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

Figure 2-6 Server and Network Architecture

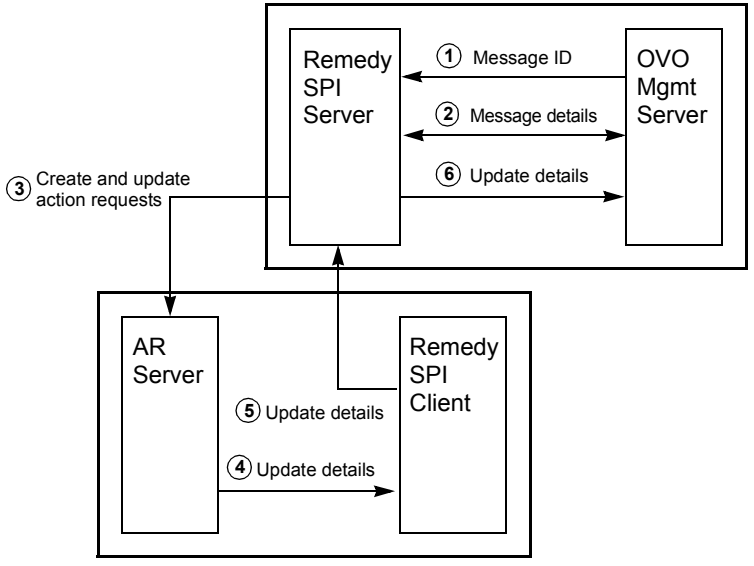
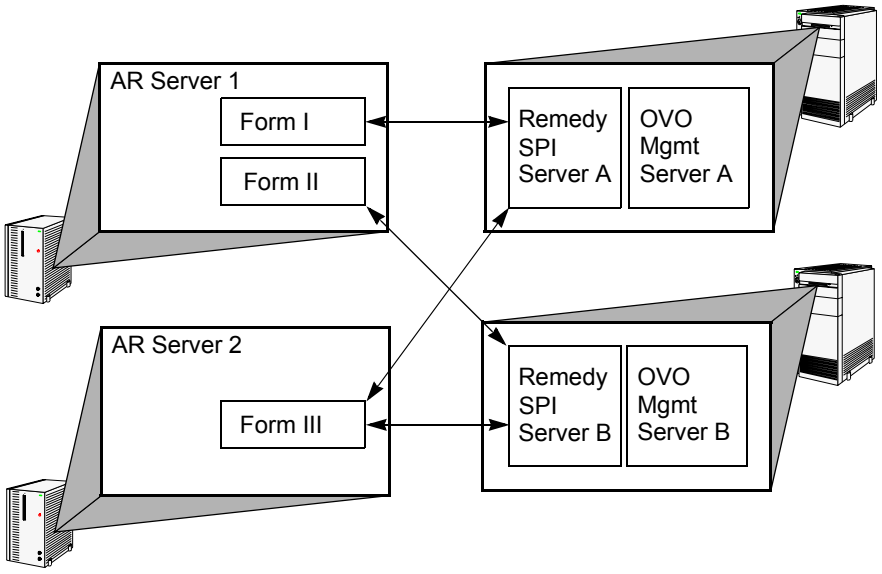


Figure 2-6 on page 27 shows that the SPI for Remedy ARS server runs on the same machine as the OVO management server. The AR server can run on the same or a different machine. The SPI for Remedy ARS client must be on the same machine as the AR server. Figure 2-6 also shows:

1. The SPI for Remedy ARS server receives the ID of a OVO message from the OVO management server via a submission program.
2. The SPI for Remedy ARS server retrieves details of the OVO message from the OVO management server via an application programmer interface (API).
3. The SPI for Remedy ARS server creates and updates action requests using the ARS API.
4. When an action request changes, the AR System calls the SPI for Remedy ARS client and passes on the details of the update via the ARS filter mechanism.
5. The SPI for Remedy ARS client sends the update details to the SPI for Remedy ARS server.
6. The SPI for Remedy ARS server updates the OVO message on the OVO management server using OVO's API.
7. When a OVO message is modified, the SPI for Remedy ARS server receives a message-change event via the OVO API.
8. OVO sends the details of the message-change event to the SPI for Remedy ARS server via the OVO API.

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

Figure 2-7 Multiple Servers



The SPI for Remedy ARS also supports multiple OVO management servers and multiple AR servers. Figure 2-7 shows that each management server has one SPI for Remedy ARS server. Each SPI for Remedy ARS server can create and update action requests on any of the AR servers. In addition, each AR server has a SPI for Remedy ARS client that can update OVO messages on any of the OVO management servers.

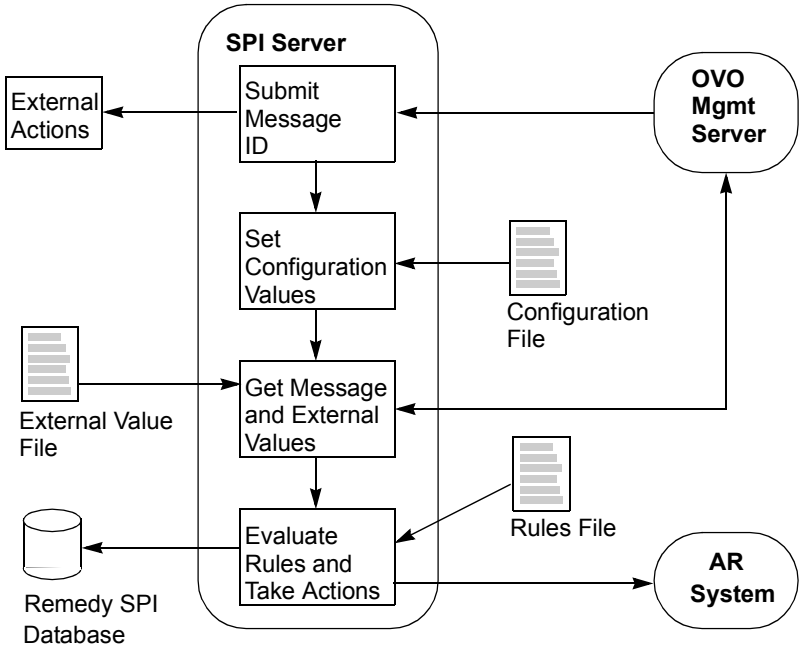
Data Components

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

- "External Actions and Data"
- "Configuration File"
- "Rules File"
- "The SPI for Remedy ARS Database"

Figure 2-8 shows how data flows in and out of the SPI for Remedy ARS server when a OVO message is submitted to the AR system.

Figure 2-8 Data Flow in the SPI for Remedy ARS



External Actions and Data

To submit a OVO message to the SPI for Remedy ARS, OVO passes the message's ID to a script called `remspisub_auto.sh/remspisub_auto.vbs` (via the Trouble Ticket Interface) and `remspisub_man.sh` (via an OVO application, OVO Unix only). 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 message does not provide. The SPI for Remedy ARS can add this extra information when it creates a new action request.

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

For more details about using external actions and external value files, see “Using External Actions and Data” on page 165.

Configuration File

The configuration file `remspi.cfg` defines the user name and passwords that the SPI for Remedy ARS uses to log in to OVO. The passwords are encrypted. For more details about creating and modifying the configuration file, see “Configuring Communication Between OVO and ARS” on page 53.

Rules File

The rules file uses OVO 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 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 SPI for Remedy ARS should update a OVO 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 messages and determine what actions need to be taken to address the problems described in the message. If a OVO message meets a condition, the SPI for Remedy ARS 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 SPI for Remedy ARS completes to create an action request. The SPI for Remedy ARS can complete fields in a schema using data from the original OVO 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 SPI for Remedy ARS Database

The SPI for Remedy ARS maintains information in the databases relating to the associations between action requests and OVO messages. Every time the SPI for Remedy ARS creates a new action request, it adds

a new record containing the OVO message ID and the action request ID. There is one database for each target in the rules file, because one OVO message may have associations with several targets.

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

SPI for Remedy ARS Components Integrated with OVO

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

- node groups for:
  - AR servers UX
  - AR servers NT
  - SPI for Remedy ARS servers
- message groups (OVO for Unix only) for:
  - AR servers
  - SPI for Remedy ARS servers
- default policies to monitor:
  - AR server processes
  - AR System log files
  - the SPI for Remedy ARS server process
  - SPI for Remedy ARS log files
  - the communication queue between OVO and the SPI
- operator and administrator users (OVO for Unix) or profiles (OVO for Windows)
- applications/tools to:
  - submit a OVO message (OVO for Unix only)
  - attach OVO messages to an existing action request (OVO for Unix only)
  - start a SPI for Remedy ARS server

— stop a SPI for Remedy ARS server
— display the SPI for Remedy ARS server status
— test the SPI for Remedy ARS
— start an AR server
— stop an AR server

3 Installing the SPI for Remedy ARS

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

In this Section

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 SPI for Remedy ARS”

NOTE The SPI for Remedy ARS is available on the *hp OpenView New and Upgraded Smart Plug-ins CD*.

Installation Prerequisites

The SPI for Remedy ARS consists of client and server components and integrates with existing OVO and AR servers. The SPI for Remedy ARS client runs on the same machine as the AR server. The SPI for Remedy ARS server runs on the same machine as the OVO server.This sectino also provides information about the following topics:

- "Disk-Space Requirements"
- "Memory (RAM) Requirements"

Table 3-1 shows the software prerequisites for the OVO for Unix servers on HP technical workstations and business servers, as well as on Sun SPARC systems.

Table 3-1 Supported OVO for Unix Servers

Product Version	Operating System				
	HP-UX		Sun Solaris		
	11.0	11.11	7	8	9
OVO 7.1+	X	X	X	X	X
OVO 8.0/8.1	X	X	X	X	X

Table 3-2 shows the software prerequisites for the OVO for Windows servers.

Table 3-2 Supported OVO for Windows Servers

Product Version	Operating System	
	Windows	
	2000 Server	2003 All variants
OVOW 7.5	X	X

NOTE The SPI for Remedy ARS is linked with AR libraries 5.1.0 on all supported OVO management server platforms. For a client program using the 5.1.0 libraries, the list of possible target AR server versions is 4.5.2, 5.0.1, 5.1.*, 6.0.*, 6.1.

Table 3-3 shows software prerequisites for the AR servers on the various supported hardware platforms.

Table 3-3 Prerequisites for the AR Server

Hardware Platform	Operating System	AR System Version			
		4.5.2	5.0.1[a]	5.1	6.0/6.1
HP Technical Workstations and Business Servers	HP-UX 10.20	-	-	-	-
	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

a. Not supported. Due to a problem in ARS 5.0.1, it is 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.

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

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

Disk-Space Requirements

Table 3-4 lists the disk space requirements for the SPI for Remedy ARS software on both the OVO management server and the AR server, which is the SPI for Remedy ARS client.

Table 3-4 Disk-Space Requirements

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

a. SPI for Remedy ARS client

Memory (RAM) Requirements

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

Installing the SPI for Remedy ARS

The installation steps differ very slightly depending on whether or not you want your AR server to be a OVO 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 SPI for Remedy ARS'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 managed nodes, please pay special attention to step 5 in the instructions below.

In order to install the SPI for Remedy ARS, 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 SPI for Remedy ARS software on the OVO management server
3. Verify that the installation of SPI for Remedy ARS software bundle completed successfully on the OVO management server
4. Assign and distribute the appropriate SPI for Remedy ARS templates to the OVO management server
5. Either:

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

      or:

   b. If you do *not* want your AR server to be a OVO managed node, copy the `itoupdate` program (`itoupdat.exe` for Windows) file from the OVO management server to the AR server manually.

      Ideally, the file should be copied to the location indicated in Table 3-10, "File Locations for the Installed SPI for Remedy ARS Client," on page 48. 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 Windows. For more information, see "ARS Filters and Active Links" on page 147.

Pre-Installation Checks - OVO for Unix only

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 SPI for Remedy ARS and explain what the contents are. Table 3-5 explains which software depot (SD) bundles are available with the HP OpenView Smart Plug-In for Remedy Action Request System integration.

Table 3-5 The SPI for Remedy ARS SD Bundles

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

Table 3-6 explains which filesets are contained in the HP OpenView Smart Plug-In for Remedy Action Request System integration product bundles.

Table 3-6 The Products in the SPI for Remedy ARS SD Bundles

SD Product	SD File Set	Description
SPI-REM-SRV	SPI-REM-SERVER	SPI for Remedy ARS server software
SPI-REM-CFG	SPI-REM-CONFIG	SPI for Remedy ARS Configuration

Table 3-6 The Products in the SPI for Remedy ARS SD Bundles

SD Product	SD File Set	Description
SPI-REM-DOC	SPI-REM-DOCUM	SPI for Remedy ARS Documentation

Table 3-7 describes the contents of the HP OpenView Smart Plug-In for Remedy Action Request System integration filesets.

Table 3-7 The SPI for Remedy ARS SD File Sets

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

Installing the SPI for Remedy ARS Software on OVO for Unix

The HP OpenView Smart Plug-In for Remedy Action Request System integration is available on the hp OpenView New and Upgraded Smart Plug-ins CD.

There are two CDs in the 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 SPI for Remedy ARS, use the first CD.

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

1. Login to the OVO 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 SPI for Remedy ARS 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 file set `OVOPC.OVOPC-UX-MGR78` (the OVO server software) to prevent installation on systems without OVO installed.

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

Installing the SPI for Remedy ARS Software on OVO for Windows

The HP OpenView Smart Plug-In for Remedy Action Request System integration is available on the *hp OpenView New and Upgraded Smart Plug-ins CD*.

Please use the appropriate installation guide for this CD-ROM to install the HP OpenView Smart Plug-In for Remedy Action Request System integration.

Verifying Installation of the SPI for Remedy ARS 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. OVO for Unix only:

   On the command line, enter: **`swlist`**

   Look for the following entries:

   ```
   SPI-Remedy B.01.10

   SPI-RemedyDoc B.01.10
   ```

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

   Node Groups

   ```
   Remedy SPI Servers

   AR Servers NT

   AR Servers UX
   ```

Message Group (OVO for Unix)	`Remedy-SPI`
	`AR System`
Application/Tool Groups	`Remedy SPI Admin`
	`Remedy SPI`
Users	`remspi_adm`
	`remspi_op`
Policy Groups	`Remedy SPI - AR Server NT`
	`Remedy SPI - AR Server UX`
	`Remedy SPI - ITO`

3. OVO for Unix only:

   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 SPI for Remedy ARS 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 3-9, “File Locations on the OVO Management Server,” on page 47.

Assigning the SPI for Remedy ARS Policies

In order to facilitate the assignment of the SPI for Remedy ARS policies to the various OVO and AR servers in your environment, you first need to add the servers to OVO and then copy the nodes to the appropriate SPI node groups, which are created during the installation of the SPI for Remedy ARS software.

Most of the following steps depend on the platform of the OVO management server. For details on how to configure OVO please consult the products manual (OVO for Windows, OVO for Unix).

1. Start the OVO GUI.

2. Add the node for the AR servers - if they are not already defined in OVO.
3. Copy the AR servers to the appropriate SPI node group, according to their platforms.
4. Copy the OVO management server to the appropriate SPI node group.

The installation of the SPI for Remedy ARS software automatically assigns the appropriate policy groups to the corresponding node group as illustrated in Table 3-8. Consequently, any managed node that you copy to the node groups listed has the correct policies assigned automatically.

Table 3-8 Automatic SPI for Remedy ARS - Policy Group Assignment

Node Group	Policy Group
AR Servers NT	SPI for Remedy ARS - AR Server NT
AR Servers UX	SPI for Remedy ARS - AR Server UX
Remedy SPI Servers	SPI for Remedy ARS - ITO

Distributing the SPI for Remedy ARS Policies (OVO for Unix)

To distribute the SPI for Remedy ARS policies to the OVO for Unix management server and, assuming it is to be a OVO 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 policy assignments you made in the previous step are correctly displayed in the `Assigned Templates` list.

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

Deploying SPI for Remedy ARS Policies (OVO for Windows)

To distribute the SPI for Remedy ARS policies to the OVO for Windows management server and, assuming it is to be a OVO managed node, the AR server, too, no additional steps are necessary.

During the installation of the SPI for Remedy ARS, the necessary policies were marked as auto deployable and thus the OVO for Windows management server will automatically distribute the policies once the nodes have been added to the SPI node groups.

Installed File Locations

The installation process copies the necessary files to the OVO management server. Table 3-9 shows the directories that the installation process creates on the OVO management server for the SPI for Remedy ARS.

We refer to the installation directory of an OVO for Windows management server as `IDIR`, e.g. `c:\Program Files\HP OpenView`

Table 3-9 File Locations on the OVO Management Server

File Type	Directory
Binaries	OVO for Unix: `/opt/OV/bin/remspi` OVO for Windows: `IDIR\bin\remspi`
Temporary and runtime	OVO for Unix: `/var/opt/OV/share/tmp/remspi` OVO for Windows: `IDIR\Data\tmp`
Log files	OVO for Unix: `/var/opt/OV/log/remspi` OVO for Windows: `IDIR\Data\log`
Configuration	OVO for Unix: `/etc/opt/OV/share/conf/remspi` OVO for Windows: `IDIR\Data\conf\remspi`

Table 3-9 File Locations on the OVO Management Server (Continued)

File Type	Directory
OVO integration	OVO for Unix: `/var/opt/OV/share/tmp/OpC_appl/remspi` OVO for Windows: `IDIR\install\remspi`
Documentation	OVO for Unix: `/opt/OV/doc` OVO for Windows: `IDIR\NLS\1033\Help`
Libraries	OVO for Unix: `/opt/OV/lib/libremspi.sl` OVO for Unix: `/opt/OV/lib/libspi.sl` OVO for Windows: `IDIR\bin\remspi`
Message Catalogs	OVO for Unix: `/opt/OV/lib/nls/C/remspi.cat` OVO for Unix: `/opt/OV/lib/nls/C/remspii.cat` OVO for Windows: `IDIR\Data\conf\remspi\remspi.cat`

Table 3-10 shows the directories for SPI for Remedy ARS clients that you install on a OVO managed node. These are the standard locations for OVO commands.

We refer to the installation directory of an OVO managed node running Windows as `IDIR`, e.g. `c:\Program Files\HP OpenView\Installed Packages\{79....}`.

Table 3-10 File Locations for the Installed SPI for Remedy ARS Client

Operating System	Directory
UNIX	`DCE agent: /var/opt/OV/bin/OpC/cmds` `HTTPS agent: /var/opt/OV/bin/instrumentation`
Windows	`IDIR`\bin\instrumentation

4 Getting Started

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

In this Section

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:

Overview

This chapter uses a simple scenario to take you through the steps required to set the communication between OVO and the ARS via the SPI for Remedy ARS. The instructions in this section assume that you have already successfully installed the Remedy SPI software on the OVO management server and assigned and distributed the Remedy SPI templates to both the OVO 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 SPI for Remedy ARS" on page 40.

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

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

1. Configure the OVO management server to communicate with your ARS server(s) via the SPI for Remedy ARS server
2. Configure the ARS server(s) to communicate with the OVO management server via the SPI for Remedy ARS 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 6, "Customizing the SPI for Remedy ARS."

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 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 and ARS work. In the process, you see how OVO 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.

NOTE

The variable `IDIR`, when used in filenames in the following text, refers to the OVO installation directory on an OVO for Windows management server.

To upload the example form 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 management server to the AR server. You can find the files on the management server in the following location:

   OVO for Unix: `/etc/opt/OV/share/conf/remspi/gui/`

   OVO for Windows: `IDIR\Data\conf\remspi\Demo`

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 and ARS

In this step, you set up and configure the connection between OVO 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 Login to OVO for the Remedy SPI" on page 53

  OVO for Unix only:

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

  — Adding a new OVO user (for the SPI for Remedy ARS) to the `User Bank` window in the OVO GUI

  — Set up the same user in the SPI for Remedy ARS-server configuration scripts

- "Configure the SPI for Remedy ARS in OVO" on page 56

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

- "Verifying the Configuration" on page 57

  Test the connection you have set up between OVO and ARS

For more information on adding and modifying users in OVO, see "SPI for Remedy ARS Users" on page 71. For information on how to set up the Trouble-Ticket (TT) interface in OVO so that OVO messages can be automatically submitted to the ARS via the SPI for Remedy ARS, see "Setting up the Trouble-Ticket Interface on OVO for UNIX" on page 91.

Create a Login to OVO for the Remedy SPI

NOTE This step is only required on an OVO for Unix management server.

To create an ARS login to OVO, you need to create a new OVO user with the appropriate permissions in OVO and copy the details of the new OVO 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 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 administator in this file, too.

NOTE

The new OVO user you create in this step is the user the Remedy SPI server will to use to access the OVO 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 SPI for Remedy ARS to forward from OVO to ARS. For more information on user privileges in OVO, see the section on the responsibility matrix in "SPI for Remedy ARS Users" on page 71 and the HP OpenView Operations Concepts Guide.

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

1. Use the OVO GUI to create a OVO user with access to the SPI for Remedy ARS applications, node groups and, in addition, the messages in the OVO database:
   a. First make a copy of an existing SPI for Remedy ARS user by opening the OVO `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 `User Bank` window and selecting the `Modify...` option from the pop-up menu.
   c. Use the OVO `Responsibility Matrix` window to ensure that the new user has access to the appropriate SPI for Remedy ARS node groups and the relevant SPI for Remedy ARS message groups.
2. Type in a password for the new OVO 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 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 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 4-1 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 SPI for Remedy ARS contains the encoded default password of the `opc_adm` user of a OVO 7.* system. This entry has to be changed for a production system.

Configure the SPI for Remedy ARS in OVO

In this section, you tell the SPI 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 via the Remedy SPI. You define these details in the `rules` file, which you can find in the following place on the OVO management server:

OVO for Unix: `/etc/opt/OV/share/conf/remspi/rules`

OVO for Windows: `IDIR\Data\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 SPI for Remedy ARS in OVO:

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 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 message generated by the condition match

4. Save the modifications to the `rules` file.
5. On the OVO management server, restart the SPI for Remedy ARS server process to activate the new configuration.

   On OVO for Unix, enter:

   ```
   ovstop RemedySPI
   ovstart RemedySPI
   ```

   On OVO for Windows, enter (or use the Windows Control Panel to restart the service):

   ```
   sc stop "HP OV SPI for Remedy Service"
   sc start "HP OV SPI for Remedy Service"
   ```

Verifying the Configuration

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

1. In the OVO GUI, start the tool `Remedy SPI Admin:SPI Test`
2. Verify that the OVO 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 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

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

- "Configuring the itoupdate Command" on page 60

  Configure the `itoupdate` command in such a way that OVO 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

- "Registering the ARS Filters" on page 61

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

- "Communicating with the AR Server through a Firewall" on page 62

  The Remedy SPI server can be instructed to use a special, fixed port to communicate with the AR server.

- "Using private queues on the AR Server" on page 62

  Define a private queue for the SPI server communication with the AR server.

- "Verifying the Configuration" on page 63

  Verify that you have successfully configured the `itoupdate` command to update OVO messages automatically as a result of changes to the status of Action Requests.

Note that `the itoupdate command` is installed on the AR server either when you install the SPI for Remedy ARS software on the AR server (the OVO managed node) or when you copy the `itoupdate` file from the OVO management server to the AR server manually. For more information, see "Installing the SPI for Remedy ARS Software on OVO for Unix" on page 42.

Configuring the itoupdate Command

This step describes how to use the `rules` file on the OVO 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 management server:

   OVO for Unix: `/etc/opt/OV/share/conf/remspi/rules`

   OVO for Windows: `IDIR\Data\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 (OVO for Unix only)

   "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 SPI for Remedy ARS software and which you installed in "Uploading the Example Form to ARS" on page 52. For more detailed information about linking changes to Action Request fields with changes to the status of OVO messages, see "Defining Update Rules" on page 100.

Registering the ARS Filters

This step describes how to register ARS filters on the OVO management server so that they can use the `itoupdate` command to update OVO 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 150

To register an ARS filter:

1. On the OVO management server, stop the SPI for Remedy ARS server process.

   On OVO for Unix, enter:

   `ovstop RemedySPI`

   On OVO for Windows, enter (or use the Windows Control Panel to stop the service):

   `sc stop "HP OV SPI for Remedy Service"`

2. Use the command `remspifilter` to register the ARS filters on the OVO management server.

   On OVO for Unix, enter:

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

   On OVO for Windows, enter:

   `IDIR\bin\remspi\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 filter. Enter:

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

4. On the OVO management server, restart the SPI for Remedy ARS server process to activate the new configuration.

   On OVO for Unix, enter:

   `ovstart RemedySPI`

On OVO for Windows, enter (or use the Windows Control Panel to start the service):

```
sc start "HP OV SPI for Remedy Service"
```

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` (OVO for Unix) or `IDIR\Data\conf\remspi\remspi.cfg` (OVO for Windows), 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*.

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` (OVO for Unix) or `IDIR\Data\conf\remspi\remspi.cfg` (OVO for Windows), 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*.

Verifying the Configuration

To verify that you have successfully configured the `itoupdate` command to update OVO 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 message in "Verifying the Configuration" on page 57. To verify that the addition or update of comments to an Action Request are reflected in the annotations attached to the original OVO message:

1. Submit a OVO message to ARS as described in "Verifying the Configuration" on page 57
2. Login to the AR server and open the `Problem Report` Schema
3. In ARS, select the Action Request created by the OVO 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, 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:

1. In ARS, select the Action Request created by the OVO 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, open the `Message Browser` window, select the message you originally submitted to ARS and ensure that the message is now owned by the appropriate SPI for Remedy ARS user.

4. In ARS, change the status of the same Action Request to `Closed`
5. In OVO, 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 SPI for Remedy ARS user.

5 Using the SPI for Remedy ARS

This section describes the SPI for Remedy ARS interface that you will use in the performance of your every-day tasks.

In this Section

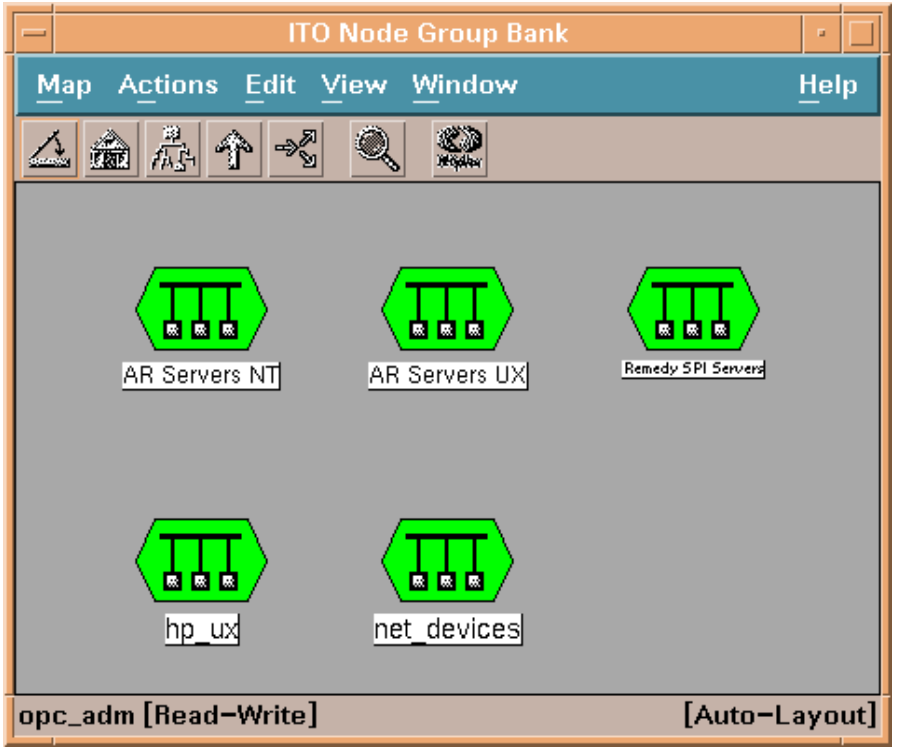
This section describes the SPI for Remedy ARS interface that you will use in the performance of your every-day tasks and gives advice and information concerning how to go about setting up and modifying the main aspects. The section covers the following areas:

SPI for Remedy ARS Node Groups

Figure 5-1 shows the node groups that are installed by default with the HP OpenView Smart Plug-In for Remedy Action Request System integration on an OVO for Unix management server. The node groups are:

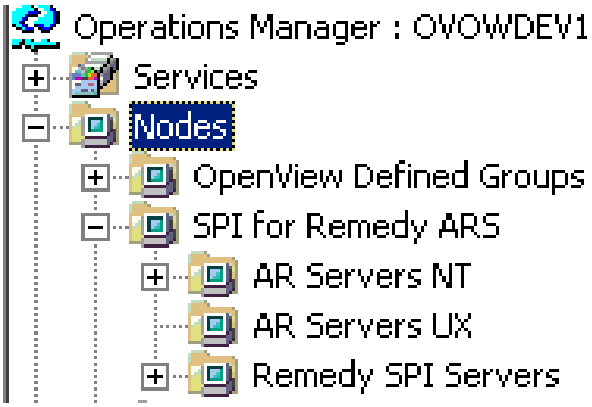
- AR Servers NT
- AR Servers UX
- Remedy SPI Servers

Figure 5-1 The SPI for Remedy ARS Node Groups (OVO for Unix)



The same node groups, but for an installation on an OVO for Windows management server are shown in Figure 5-2 on page 68.

Figure 5-2 The SPI for Remedy ARS Node Groups (OVO for Windows)



The AR servers in your ARS environment should be added to the `AR Servers NT` and `AR Servers UX` node group as appropriate.

The system running the SPI for Remedy ARS server, normally the OVO management server, should be added to the node group `Remedy SPI Servers` in the same way.

The installation of the SPI for Remedy ARS software automatically assigns the appropriate policy group to the corresponding node group as illustrated in Table 3-8 on page 46. Consequently, the managed nodes that you copy to the node groups listed have the correct policies 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 37.

You can then assign the message groups `AR System` and `Remedy-SPI` to the SPI for Remedy ARS users you add and make them responsible for your AR systems and the SPI for Remedy ARS in general. In this way, the messages generated by SPI for Remedy ARS systems are sent automatically to the SPI for Remedy ARS users whom you make directly responsible for these systems. For more information about SPI for Remedy ARS users, see “SPI for Remedy ARS Users” on page 71. For more information about SPI for Remedy ARS message groups, see “SPI for Remedy ARS Message Groups” on page 69.

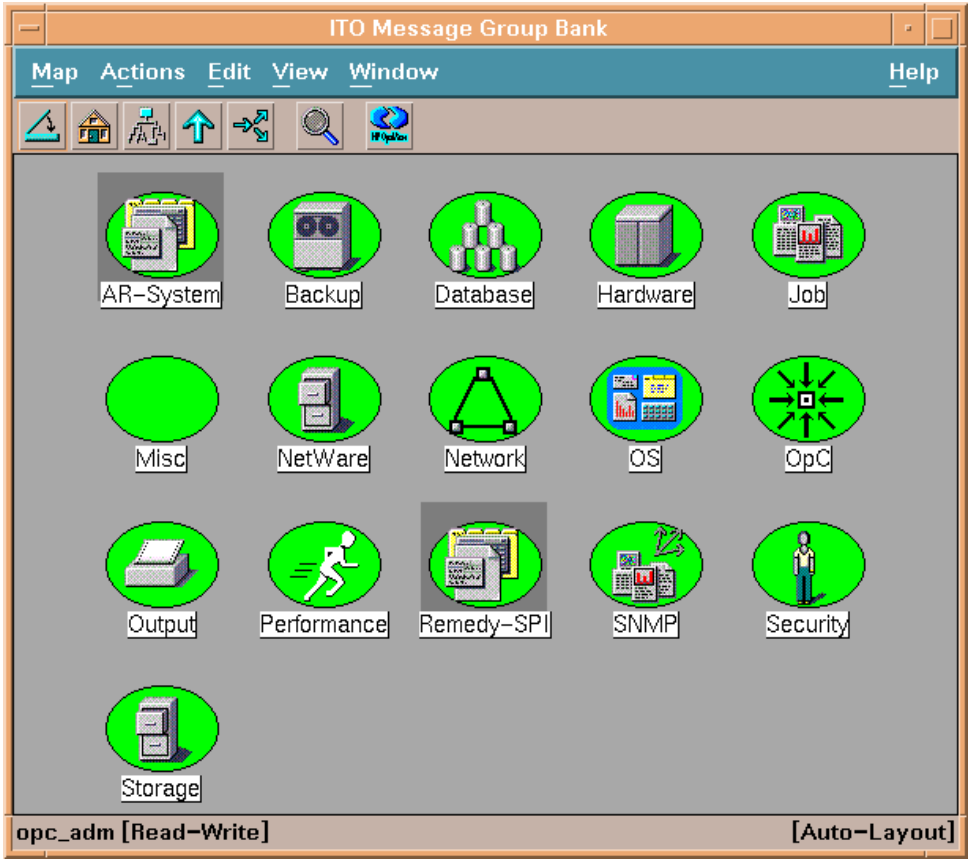
SPI for Remedy ARS Message Groups

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 SPI for Remedy ARS users whom you add and make responsible for your AR systems and the SPI for Remedy ARS in general.

Figure 5-3 The SPI for Remedy ARS Message Groups (OVO for Unix)



For more information about SPI for Remedy ARS users, see “SPI for Remedy ARS Users” on page 71. For more information about SPI for Remedy ARS node groups, see “SPI for Remedy ARS Node Groups” on page 67.

SPI for Remedy ARS Users

This section describes the default SPI for Remedy ARS 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 5-4 **The SPI for Remedy ARS User Bank (OVO for Unix)**

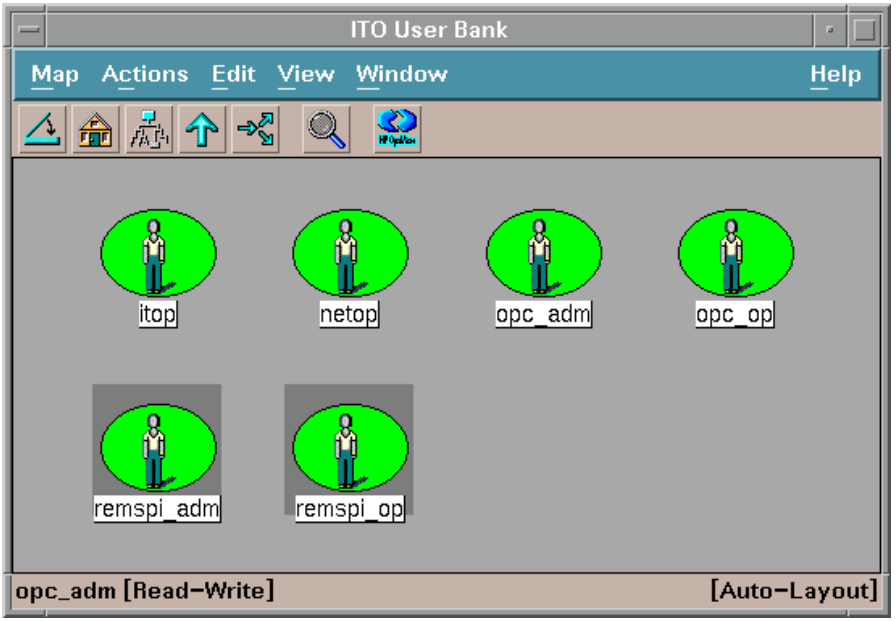
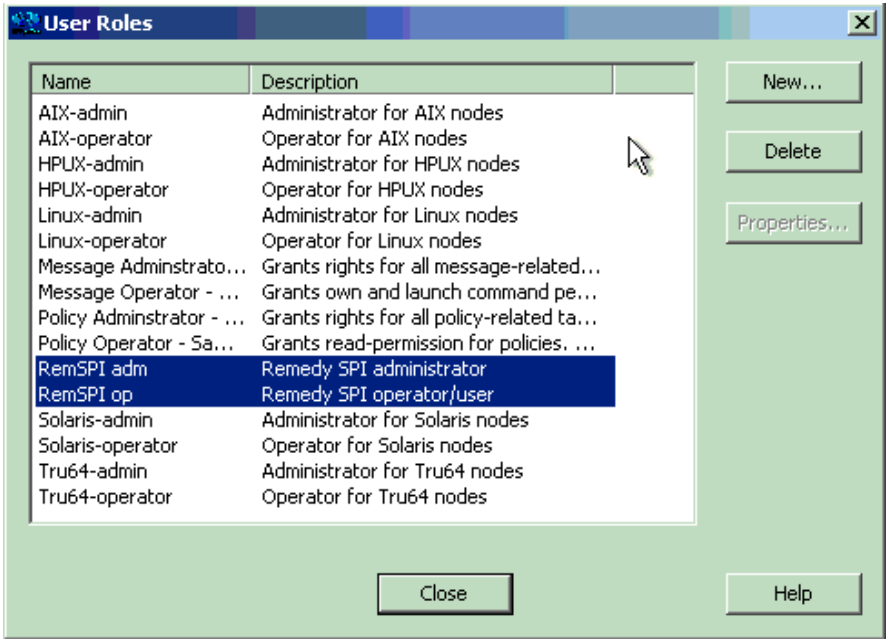


Figure 5-5 **The SPI for Remedy ARS User Bank (OVO for Windows)**

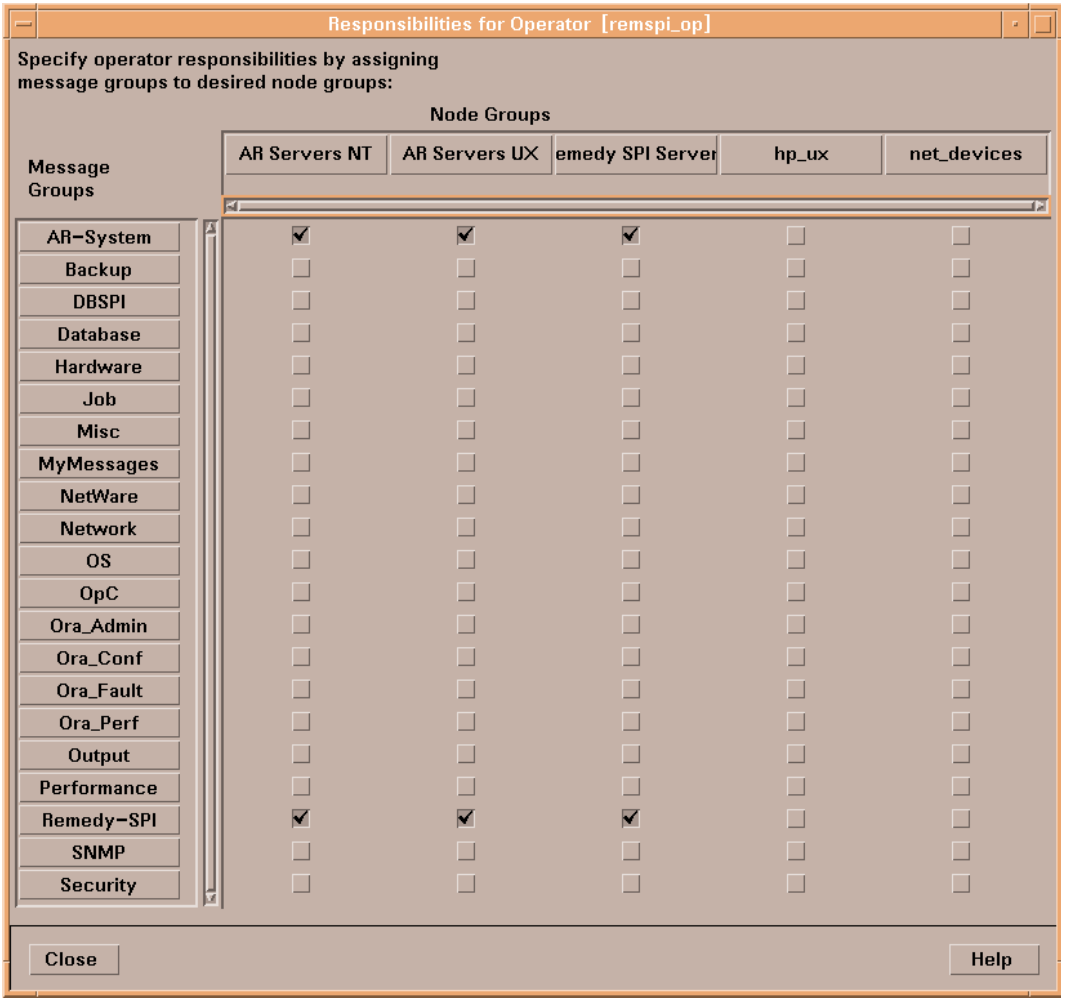


The default OVO user accounts for the SPI for Remedy ARS, `rempsi_adm` and `rempsi_op` are intended as examples: they can be used as a basis for the creation of “real” OVO user accounts, which are added by the OVO administrator. The default SPI for Remedy ARS user accounts have access to the default SPI for Remedy ARS applications in order to allow them to manage the SPI for Remedy ARS (starting and stopping AR servers, for example) and to work with the SPI for Remedy ARS (submitting and attaching messages). In addition, the default SPI for Remedy ARS users can “see” the default SPI for Remedy ARS 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 5-6 on page 73.

The OVO account you set up for the SPI for Remedy ARS server in "Configuring Communication Between OVO and ARS" on page 53 and which is defined in the configuration file `remspi.cfg` is used by the SPI for Remedy ARS server to access messages in the OVO database. Clearly, this account must be granted access to all the messages which should be forwarded to the AR system.

Figure 5-6 SPI for Remedy ARS User's Responsibilities (OVO for Unix)



You need to pay particular attention to which node groups are assigned to which message groups. In the example illustrated in Figure 5-6, SPI for Remedy ARS User's Responsibilities (OVO for Unix), the SPI for Remedy ARS 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 SPI for Remedy ARS, see "SPI for Remedy ARS Node Groups" on page 67 and "SPI for Remedy ARS Message Groups" on page 69.

SPI for Remedy ARS Tool Groups

To assist the SPI for Remedy ARS 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 tool groups specifically for the SPI for Remedy ARS. The new tool groups are:

- Remedy SPI

  contains tools for operating the SPI for Remedy ARS

- Remedy SPI Admin

  contains tools for the administration of the SPI for Remedy ARS

Figure 5-7 shows the two SPI for Remedy ARS-specific application groups in the `OVO Application Bank` window on an OVO for Unix management server.

Figure 5-7 SPI for Remedy ARS Application Groups (OVO for Unix)

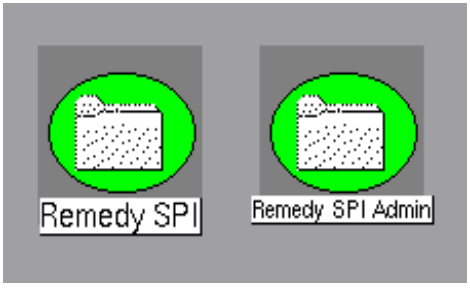
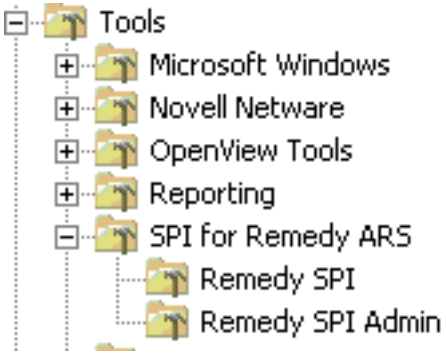


Figure 5-8 on page 76 shows the two SPI for Remedy ARS specific tool groups in an OVO for Windows management server.

Figure 5-8 SPI for Remedy ARS Application Groups (OVO for Windows)



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

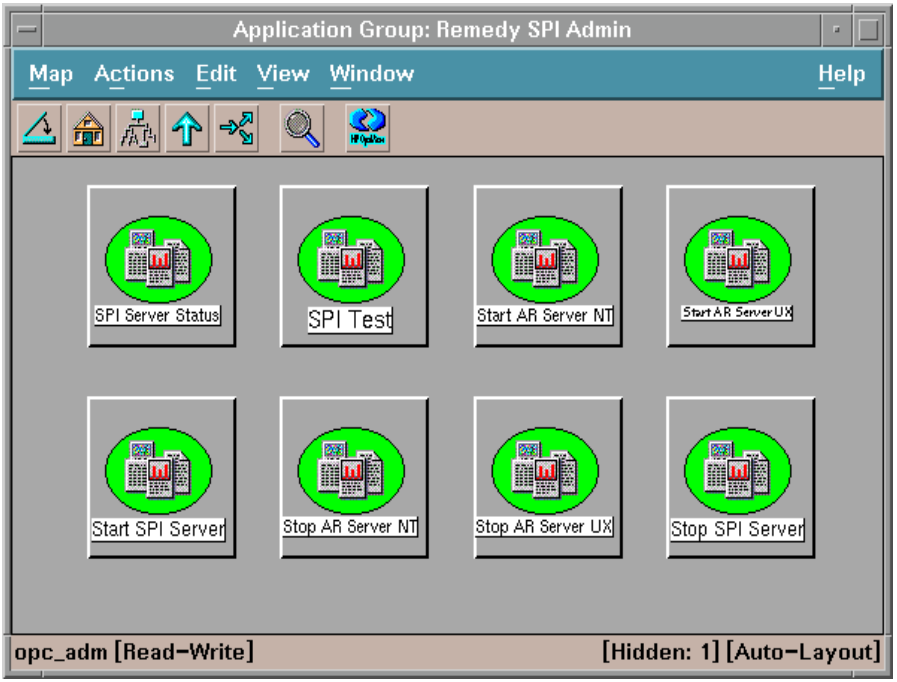
For more information about the contents of the `Remedy SPI` and `Remedy SPI Admin` tool groups, see "SPI for Remedy ARS Operator Tools" on page 81 and "SPI for Remedy ARS Administrator Tools" on page 77, respectively.

SPI for Remedy ARS Administrator Tools

The tool group `Remedy SPI Admin` contains tools that enable the SPI for Remedy ARS administrator to manage the SPI for Remedy ARS from within OVO. Figure 5-9 shows the `Remedy SPI Admin` tool group, which by default contains the following tools:

- "The SPI for Remedy ARS Server's Status" on page 78
- "Starting and Stopping the SPI for Remedy ARS Server" on page 79
- "Testing the SPI for Remedy ARS" on page 80
- "Starting and Stopping the AR Server" on page 80

Figure 5-9 Administrator Applications (OVO for Unix)



NOTE

`Remedy SPI Admin` is the default tool group that the installation procedure creates for a SPI for Remedy ARS administrator. You may add, modify, or group the tools according to the requirements of your organization.

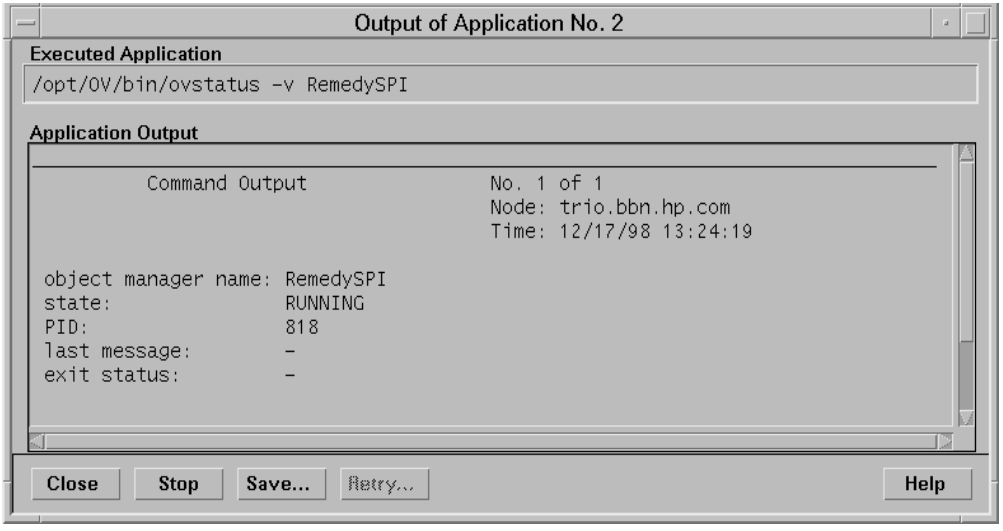
The same tools are available on an OVO for Windows management server.

The SPI for Remedy ARS Server's Status

You can check the status of the SPI for Remedy ARS server by starting the `SPI Server Status` tool. The status information appears in an `Output of Application` window.

Figure 5-10 shows example status information that appears when a SPI for Remedy ARS server is running normally on an OVO for Unix management server.

Figure 5-10 SPI for Remedy ARS Server Status



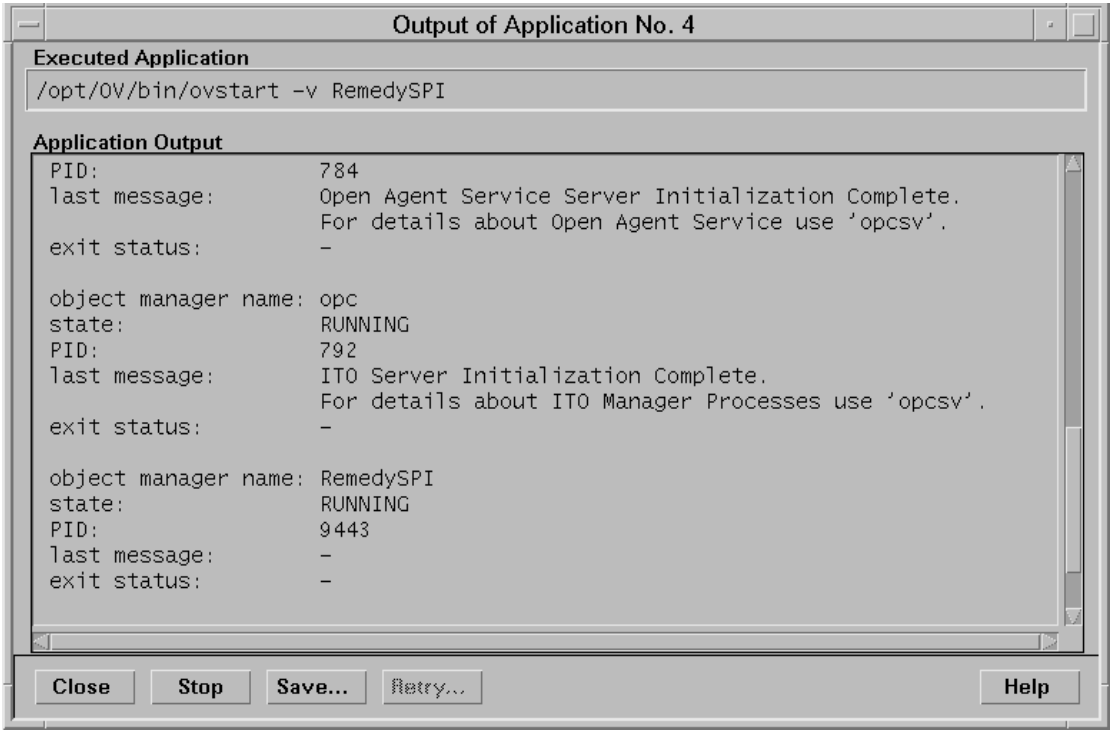
Starting and Stopping the SPI for Remedy ARS Server

You can start or stop the SPI for Remedy ARS server by starting the `Start SPI Server` or `Stop SPI Server` tools. The tool confirms the new status of the SPI for Remedy ARS server in an `Output of Application` window.

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

Figure 5-11 shows example output that appears when a SPI for Remedy ARS server starts successfully on an OVO for Unix management server.

Figure 5-11 SPI for Remedy ARS Server Start-up



Testing the SPI for Remedy ARS

As an administrator you can test the status of the Remedy SPI using the `SPI Test` tool . A message appears in the OVO `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 OVO managed node. To do this:

1. Select the managed node in the OVO GUI.
2. Start the `Start AR Server` or `Stop AR Server` tool.

SPI for Remedy ARS Operator Tools

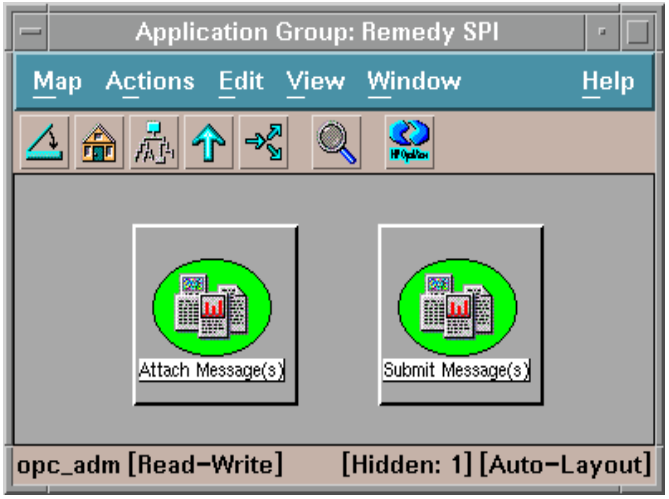
NOTE The tools described in this section are available only on an OVO for Unix management server

The tool group `Remedy SPI` contains tools that enable the SPI for Remedy ARS operate to carry out actions from within OVO. Figure 5-12 on page 81 shows this tool group, which by default contains the following tools :

- "Manually Submitting Messages to an AR System" on page 82
- "Attaching Messages to Existing Action Requests" on page 83

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

Figure 5-12 Operator Applications



Manually Submitting Messages to an AR System

You can use OVO 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, ARS, and the SPI for Remedy ARS.

To manually submit OVO 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:

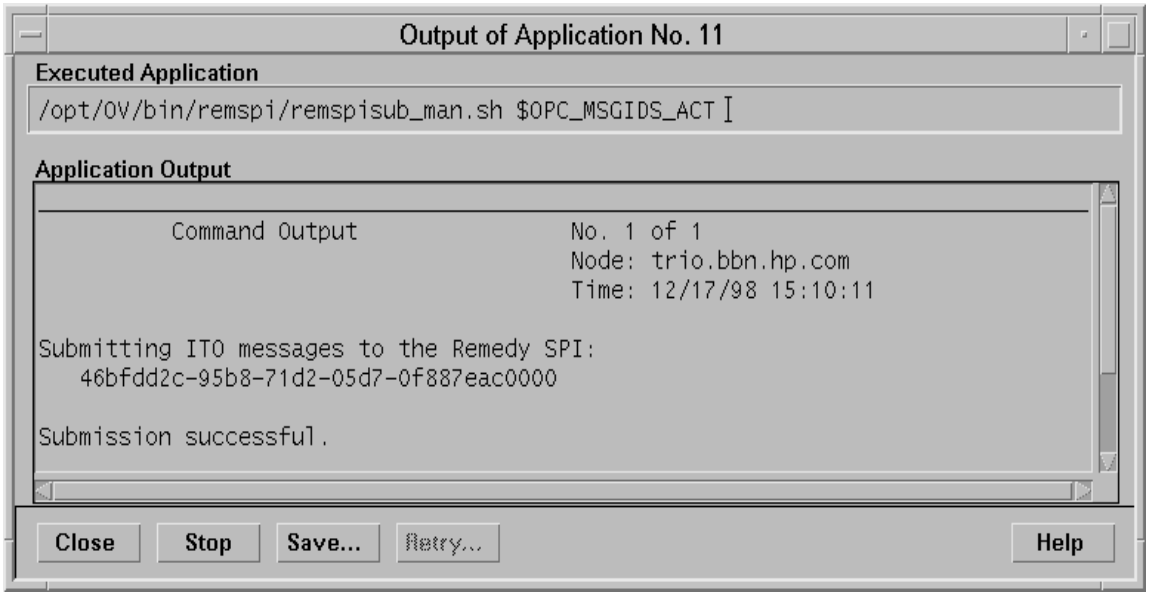
- The OVO message does not match any of the conditions in the rules file. For more details, see "Defining Conditions" on page 115.
- OVO or an OVO user has already submitted the same OVO message. You cannot resubmit OVO messages. The SPI for Remedy ARS updates them automatically according to the update 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 SPI for Remedy ARS evaluates conditions for the message with the highest severity first. If more than one message has the same severity, the SPI for Remedy ARS 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 5-13 shows an example of the output that appears when a user submits an OVO message successfully.

Figure 5-13 Manual Message Submission



Attaching Messages to Existing Action Requests

You can attach one or more new OVO 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 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 SPI for Remedy ARS treats messages attached to an action request as an update to the action request. For more details, see “Defining Update Rules” on page 100.

If you want to configure the SPI for Remedy ARS to be able to perform attachments to existing action request automatically, see “Configuring automatic attachments” on page 135

6 Customizing the SPI for Remedy ARS

This section describes how to customize the HP OpenView Smart Plug-in for Remedy ARS to suit the demands of your environment.

In this Section

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

Customizing OVO

The SPI for Remedy ARS's installation process creates a default configuration within OVO. You may need to change these, depending on your organization's needs. In this section, you can find out how to perform the following actions:

Customizing OVO GUI Objects

After you install the SPI for Remedy ARS, new objects appear in the OVO GUI that enable you to work with the SPI for Remedy ARS. Table 6-1 shows a list of these objects.

Table 6-1 SPI for Remedy ARS Objects in the OVO GUI

Object Type	Name	Description
Node Group	SPI for Remedy ARS Servers	Nodes on which a SPI for Remedy ARS server runs. By default, this contains the current SPI for Remedy ARS 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.
Message Group	Remedy-SPI	OVO messages that result from monitoring the SPI for Remedy ARS server
	AR-System	OVO messages that result from monitoring an AR server.

Table 6-1 SPI for Remedy ARS Objects in the OVO GUI (Continued)

Object Type	Name	Description
Application/Tool Group	Remedy SPI Admin	Applications that the SPI for Remedy ARS administrator uses to manage the SPI for Remedy ARS and AR servers.
	Remedy SPI	Applications that SPI for Remedy ARS operators and administrators use to work with the SPI for Remedy ARS.
User (OVO for UNIX) User Roles (OVO for Windows)	remspi_adm RemSPI adm	The SPI for Remedy ARS administrator has access to all SPI for Remedy ARS node, message, and application groups. After installation, the default password is `RemspI_adm`.
	remspi_op RemSPI op	The SPI for Remedy ARS operator has access to all SPI for Remedy ARS node and message groups, but only to the SPI for Remedy ARS application group. After installation, the default password is `RemspI_op`.

Modifying the SPI for Remedy ARS Policies

You can modify the policies supplied with the SPI for Remedy ARS using the `Message Source Templates` window in the OVO/UNIX GUI or the `Policy management / Policy groups / SPI for Remedy ARS` tree in the OVO/Windows GUI. For example, you might want to modify the configuration of a policy in order to change the polling interval or set up the Trouble-Ticket interface for individual message conditions.

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

To change or modify a OVO for UNIX template:

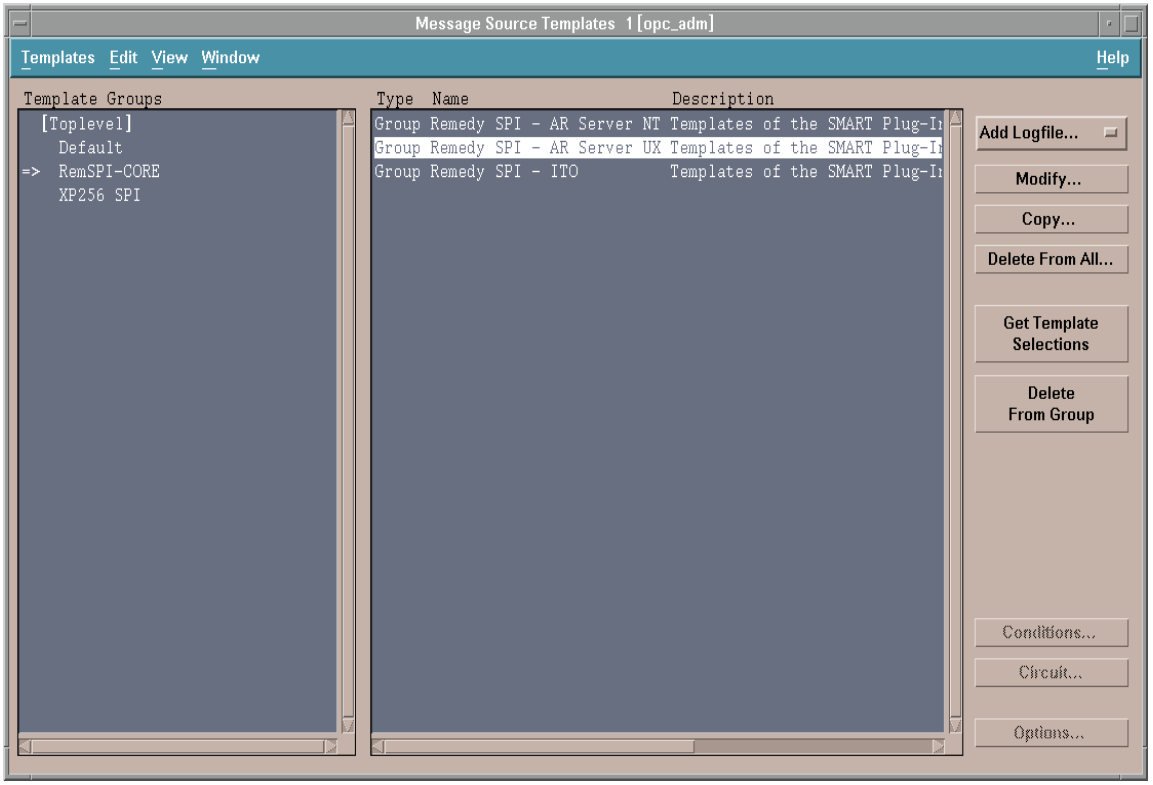
1. Open the `Message Source Templates` window shown in Figure 6-1 on page 91
2. Expand the template group containing the templates you want to modify
3. Select the appropriate template, and click the `Modify...` button
4. Save the changes you have made and re-distribute the templates to the appropriate nodes

To change or modify a OVO for Windows policy:

1. Open the `Policy management / Policy groups / SPI for Remedy ARS` subtree.
2. Expand the policy group containing the policies you want to modify
3. Select the appropriate policy, and double-click the policy name

4. Save the changes you have made

Figure 6-1 The Message Source Templates Window (OVO for UNIX)



Setting up the Trouble-Ticket Interface on OVO for UNIX

This section describes how to set up the Trouble-Ticket (TT) interface in OVO for UNIX so that messages generated by OVO for UNIX can be submitted automatically to the AR server via SPI for Remedy ARS. In addition, you can also choose to activate the Trouble-Ticket interface for individual message conditions - so that OVO for 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 for UNIX management server:

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

   Actions:Utilities-> Trouble Ticket...

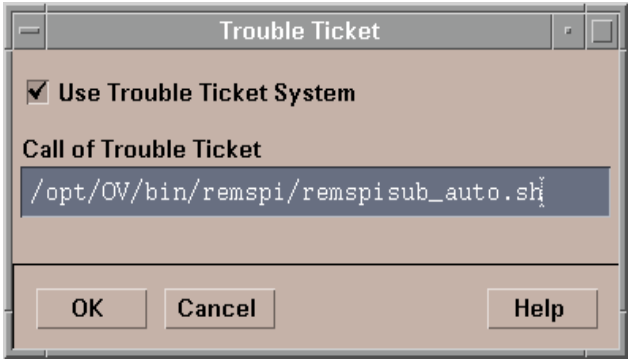
b. In the Trouble Ticket window that appears:

   - Check the option: Use Trouble Ticket Interface
   - Enter the path to the SPI for Remedy ARS 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 6-2 The Trouble-Ticket Window in OVO for UNIX



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

   a. In the OVO 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.

Setting up the Trouble-Ticket Interface on OVO for Windows

This section describes how to set up the Trouble-Ticket (TT) interface in OVO for Windows so that messages generated by OVO for Windows can be submitted automatically to the AR server via the SPI for Remedy ARS. In addition, you can also choose to activate the Trouble-Ticket interface for individual message conditions - so that OVO for Windows messages generated as a result of matching specific template conditions are copied to the TT interface.

The `Windows Management Interface` type policy `Remedy SPI - ITO / SubmitTTtoRemSPI` is deployed automatically to the OVO for Windows management server, if the server is assigned to the node group `Remedy SPI Servers`. This policy catches all newly created messages and forwards them to the SPI for Remedy ARS, which then checks whether this message has been created by a policy message condition with the `Forward to Trouble Ticket` flag set to *true*.

If this is the case, then an AR ticket is created according to the configuration of the SPI for Remedy ARS.

NOTE You do not have to do anythinghi to enable the Trouble Ticket interface on the OVO for Windows management server.

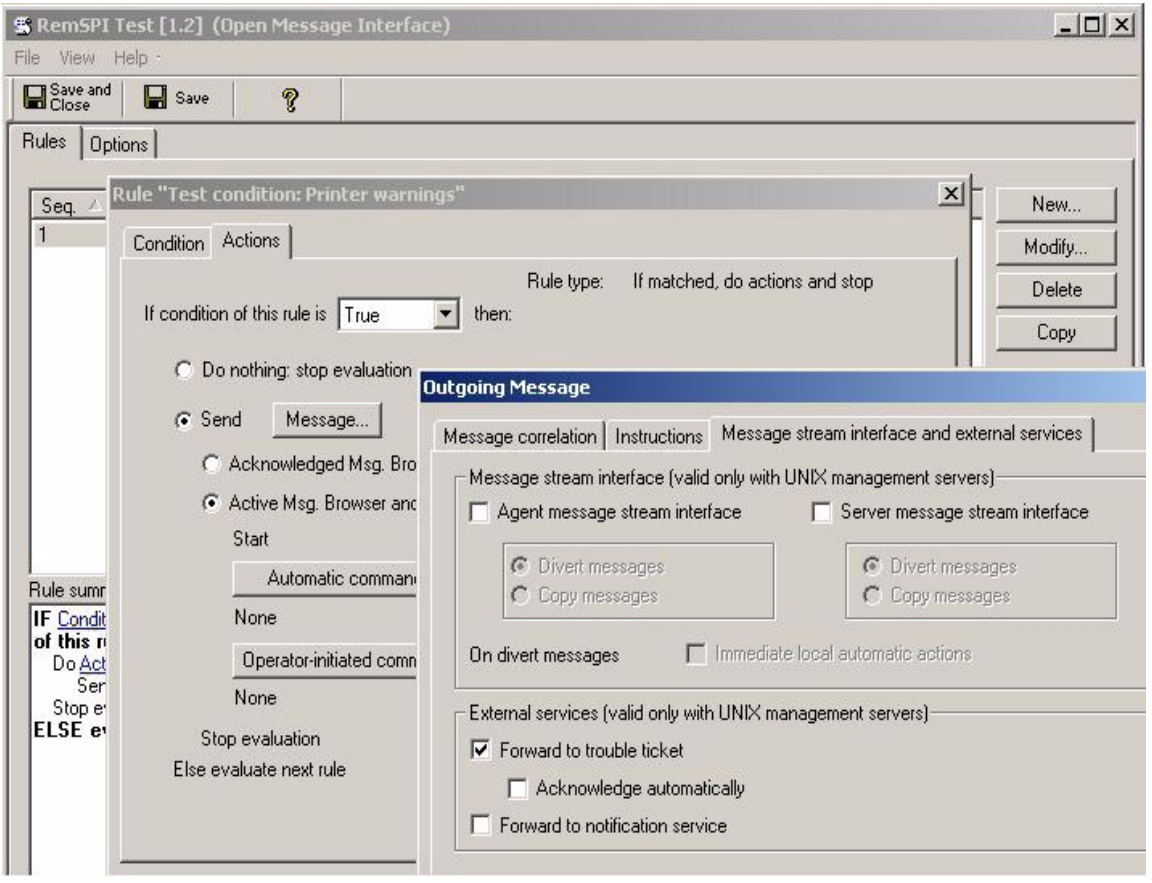
To enable specific messages to be sent to this Trouble Ticket interface - and thus to be forwarded to the AR server - the following steps may be used.

1. Activate the Trouble-Ticket interface for the individual OVO for Windows message conditions that will generate messages for ARS:
   a. In the OVO for Windows policy editor window, select the conditions you want to modify.
   b. Click the `Modify...` button
   c. In the `Actions` section of the `Rule` window, click the `Message` button.
   d. In the `Outgoing Message` window, go to the `Message stream interface and external services` section and check the `Forward to trouble ticket` option. Don't mind that this option

is flagged as valid for OVO for UNIX management servers only. This option will be evaluated by the SPI for Remedy ARS, not the OVO for Windows management server.

e. Close all open windows with OK and save the policy. If the policy is not automatically deployed, redeploy the policy to the managed node.

Figure 6-3 Activating automatic forwarding to the Trouble Ticket interface in OVO for Windows



Customizing the Rules File

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

NOTE After making changes to the `rules` file, you need to apply the changes by saving the file and stopping and restarting the SPI for Remedy ARS server process, which runs on the OVO 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 SPI for Remedy ARS
- targets that describe AR servers and schemas
- update rules that determine how the SPI for Remedy ARS updates an OVO message if the corresponding action request changes
- submit rules that determine how the SPI for Remedy ARS updates an OVO message if the message is submitted as an AR ticket

- conditions that filter the OVO messages submitted to the SPI for Remedy ARS
- actions that determine how to create or update action requests when a OVO message satisfies a defined condition

Figure 6-4 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 6-4, 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 SPI for Remedy ARS 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 6-14 on page 138.

Defining Global Options

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

Table 6-2 Global Options

Option	Description
NO_AR_UPDATE	The SPI for Remedy ARS 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 6-2 Global Options (Continued)

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 6-3 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
```

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 SPI for Remedy ARS messages are generated when buffering starts and ends.

Change events for buffered OVO messages are not buffered.

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 SPI for Remedy ARS. This information enables the SPI for Remedy ARS server to connect to the AR server and create or change action requests that belong to a particular schema.

Table 6-3 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.*
USER	-	The user name with which the SPI for Remedy ARS 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 6-3 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 SPI for Remedy ARS must use.

The following example defines a target for the form `Problem Report` illustrated in Figure 2-3 on page 20, 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 SPI for Remedy ARS server updates the associated OVO 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 6-4 on page 101 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 (OVO for UNIX only)"
- "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*]

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 page 153.

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 page 153..

Table 6-4 Update Keywords

Keyword	Parameters	Description
ACKNOWLEDGE	[IF_MODIFIED] *field ID* *value*	Use this keyword to acknowledge the OVO message if the *field ID* has the given *value*. This also removes the association between the action request and the OVO message from the SPI for Remedy ARS 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.

Table 6-4 Update Keywords (Continued)

Keyword	Parameters	Description
UNACKNOWLEDGE	[IF_MODIFIED] *field ID* *value*	Use this keyword to unacknowledge the OVO message if the *field ID* has the given *value*. This action is valid only if the message was locally acknowledgen using the OVO 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.
ANNOTATE	[IF_MODIFIED *field ID*] *format*	Use this keyword to add an annotation to the OVO message. The *format* desccribes 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.

Table 6-4 Update Keywords (Continued)

Keyword	Parameters	Description
ANNOTATE	[IF_MODIFIED] *<field ID* *value>* *format*	Use this keyword to add an annotation to the OVO 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 (OVO for UNIX only)	*command name*	Use this keyword to escalate the OVO message if the SPI for Remedy ARS client passes the given *command name*.
OP_ACTION	*command name*	Use this keyword to start an operator-initiated action if the SPI for Remedy ARS client passes the given *command name*.
OWN	[IF_MODIFIED] *field ID* *value*	Use this keyword to take ownership of the OVO 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.

Table 6-4 Update Keywords (Continued)

Keyword	Parameters	Description
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 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 6-4 Update Keywords (Continued)

Keyword	Parameters	Description
DISOWN	[IF_MODIFIED] *field ID* *value*	Use this keyword to remove ownership of the OVO 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 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.

Acknowledging Messages

The following example shows an update rule that acknowledges the OVO 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 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 contructed 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 page 153..

For information on the problems that can occur when acknowledging a large number of messages at the same time, see "Troubleshooting SPI for Remedy ARS Usage" on page 181

Unacknowledging Messages

The following example shows an update rule that unacknowledges the OVO 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 SPI for Remedy ARS 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 contructed 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 6-5 shows the annotation variables that you can use.

Table 6-5 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 (OVO for UNIX only)

The next example shows an update rule that escalates a OVO for UNIX message if an appropriate OVO for UNIX escalation manager exists. The command name (here `escalate`) must match the command parameter in the active link that calls the SPI for Remedy ARS client. For more details about setting up an active link, see "Configuring an Active Link" on page 155.

```
ITO_UPDATE
  ESCALATE "escalate"
```

Starting an Operator-Initiated Action

The next example shows an update rule that starts the OVO 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 SPI for Remedy ARS client. For more details about setting up an active link, see"Configuring an Active Link" on page 155.

```
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 SPI for Remedy ARS to take ownership of the corresponding OVO 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 SPI for Remedy ARS to take ownership of the OVO 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 user that you specify in the configuration file `remspi.cfg` becomes the owner of the OVO messages.

For information on the problems that can occur when taking ownership of a large number of messages at the same time, see "Troubleshooting SPI for Remedy ARS Usage" on page 181

Removing Ownership from a Message

The following example shows an update rules that use the `DISOWN` keyword. The rule instructs the SPI for Remedy ARS to remove ownership from the corresponding OVO 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 SPI for Remedy ARS server updates the OVO 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.

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"

You then assign keywords with parameters to the ON_SUBMIT attribute. Table 6-6 on page 109 shows the keywords you use to define for a target.

Table 6-6 Submit Keywords

Keyword	Parameters	Description
ACKNOWLEDGE	-	Use this keyword to acknowledge the OVO message right after the message is submitted as an AR ticket. This also removes the association between the action request and the OVO message from the SPI for Remedy ARS database.
OWN	-	Use this keyword to take ownership of the OVO message right after the message is submitted as an AR ticket.

Table 6-6 Submit Keywords (Continued)

Keyword	Parameters	Description
FORCE OWN	-	Use this keyword to remove the ownership of the message and then to take ownership of the OVO 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.
DISOWN	-	Use this keyword to remove ownership of the OVO message right after the message is submitted as an AR ticket. Note that in order to use this keyword, the password for the OVO 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 message text (if specified).

Table 6-6 Submit Keywords (Continued)

Keyword	Parameters	Description
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 message right after the message is submitted as an AR ticket. Note that after the modification the message is automatically owned by the OVO 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 core system.
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 message which is generated right after the current message is sucessfully 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"

Table 6-6 Submit Keywords (Continued)

Keyword	Parameters	Description
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 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 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 SPI for Remedy ARS Usage" on page 181

Taking Ownership of a Message

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

The rule instructs the SPI for Remedy ARS to take ownership of the OVO 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 user that you specify in the configuration file `remspi.cfg` becomes the owner of the OVO messages.

If you use the `FORCE` keyword, you have to configure the password of the OVO administrator in the `remspi.cfg`configuration 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 SPI for Remedy ARS Usage" on page 181

Removing Ownership from a Message

The following example shows a submit rule that uses the `DISOWN` keyword. The rule instructs the SPI for Remedy ARS to remove ownership from the corresponding OVO 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 administrator in the `remspi.cfg`configuration 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 SPI for Remedy ARS to prepend the ticket of the constructed AR ticket to the text of the corresponding OVO 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 SPI for Remedy ARS 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 SPI for Remedy ARS 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 sends to the SPI for Remedy ARS 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 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 SPI for Remedy ARS evaluates conditions in the order that they appear in the rules file. If a OVO message satisfies a condition, the SPI for Remedy ARS 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 message. The keyword `EXTERNAL` in the test part of a condition refers to an external value file, not to a message attribute.

Table 6-7 Condition Keywords

Keyword	Description
AA_STATUS	The status of the automatic action of the OVO message. Syntax: `AA_STATUS [NOT] status` where *status* one of UNDEFINED, AVAILABLE, STARTED, FINISHED, FAILED

Table 6-7 Condition Keywords (Continued)

Keyword	Description
APPLICATION	The application string attached to the OVO message. Syntax: `APPLICATION [NOT] [LIKE] string`
CMA (OVO for UNIX only)	This keyword needs two strings as parameters. The first parameter is the name of an OVO Custom Message Attribute, the second parameter the exepected value of this CMA. This condition term is true if the OVO message has a Custom Message Attribute with that name and value. Syntax: `CMA key-string [NOT] [LIKE] val-string`
DUPLICATES (OVO for UNIX only)	The number of duplicates of the OVO message. Syntax: `DUPLICATES op number` where op one of `==, !=, >=, <=, >, <`
EXTERNAL	An value in an external value file. For more details, see "Referencing External Values" on page 169. Note that this is keyword behaves different than others in this list. This keyword does not refer to an attribute of the OVO message, but to a value in an external file. Syntax: `EXTERNAL key-string [NOT] [LIKE] val-str`
ITO_SUBMITTER (OVO for UNIX only)	The name of the OVO user who submitted the message. Syntax: `ITO_SUBMITTER [NOT] [LIKE] string`

Table 6-7 Condition Keywords (Continued)

Keyword	Description
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).
NODE_GROUP (OVO for UNIX only)	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`
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

Table 6-7 Condition Keywords (Continued)

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

Table 6-7 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 message match the values you give for the keywords, the OVO 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 message.

The next example shows a suppress condition that filters out all OVO messages that have the severity `Normal`. From the OVO 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 SPI for Remedy ARS has already used, the SPI for Remedy ARS 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`.

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 6-8 on page 120 lists some examples for using regular expressions in conditions.

Table 6-8 Regular Expressions in Message Conditions

Syntax	Description
`OBJECT LIKE "^oracle-[0-9]+$"`	Matches if the OVO message object attribut starts with the string `oracle-` and ends with a non empty sequence of digits.
`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 SPI for Remedy ARS. The section covers the following areas:

- "Using Action Variables"
- "Defining Values using Composite Formats"
- "Setting Default Values"
- "Setting Enumerated Field Values"
- "Handling Multiple OVO 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 SPI for Remedy ARS creates or updates an action request when a OVO message satisfies the condition. To define an action, you specify the target and state how the SPI for Remedy ARS 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 `SET TARGET` 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."

  SET 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 SPI for Remedy ARS nonetheless attempts to convert in the appropriate manner. The SPI for Remedy ARS truncates string values if necessary.

Using Action Variables

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

Table 6-9 shows the action variables that you can use. Each time you use an action variable, place $ before and after the variable name.

Table 6-9 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 user who acknowledged the message.	Yes
ANNO_CNT	The number of annotations to this message.	No
ANNO_USER	The name of the OVO user who added an annotation last.	Yes

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
DUPLICATES (OVO for UNIX only)	The number of duplicates of this OVO message.	No
ESCAL_SRV (OVO for UNIX only)	The name of the server from or to which the message is escalated.	Yes
ESCAL_STAT (OVO for UNIX only)	Whether the message is escalated to or from another management server, or not at all.	No
EXTERNAL	This action variable needs the name of a key in the external file as a parameter. This pair is tehn 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

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
HLP_TEXT	The instructions associated with the message. If OVO has an external instruction interface for this message, the SPI for Remedy ARS 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 machine where the OVO management server and SPI for Remedy ARS server run.	No
INSTR_IF (OVO for UNIX only)	The name of any instruction interface for this message.	Yes
INSTR_PAR (OVO for UNIX only)	Details of any instruction interface parameters.	Yes
INSTR_TYPE	Whether an instruction for this message is internal or uses an interface.	Yes

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
ITO_MSI (OVO for UNIX only)	Whether OVO passed the message to the message stream interface.	No
ITO_NOTIFY	Whether OVO passed the message to a notification interface.	No
ITO_SUBMITTER (OVOfor UNIX only)	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 passed the message to the trouble ticket interface.	No
ITO_TT_ACK	Whether OVO 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 agent.	No
ITO_USER	The OVO user name which the SPI for Remedy ARS used to log in to OVO.	No
LOG_ONLY	Whether OVO immediately moves the message to the history database.	No

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
MSG_ANNO	The annotations added to the message. When the SPI for Remedy ARS creates an action request, it combines all the existing annotations. When the SPI for Remedy ARS 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 managed node generated the message.	No

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
NODE_NAME	The name of the machine that sent the message.	No
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 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 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 management server received the message.	No

Table 6-9 Action Variables (Continued)

Variable	Description	Can Be Empty ?
SRV_TIME	The time at which the OVO management server received the message.	No

Table 6-9 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 message into field `1000000003`
- the name of the application that generated the OVO 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 message. You do this using composite formats. Table 6-10 shows the composite formats available. The composite format MSG_EVENT varies automatically, depending on how the OVO message changed. Table 6-11 shows the formats that can apply for MSG_EVENT.

Table 6-10 Composite Formats

Format Name	Description
MSG_EVENT	A format that varies automatically, depending on the type of change made to the OVO message. Table 6-11 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 6-11 MSG_EVENT Formats

Event	Format
MSG_EVT_AA_END	"Automatic action of OVO message MSG_ID finished."
MSG_EVT_AA_START	"Automatic action of OVO 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."

Table 6-11 MSG_EVENT Formats (Continued)

Event	Format
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 $ESCAL_SRV$."
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 6-9 may contain no value, depending on the OVO 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.

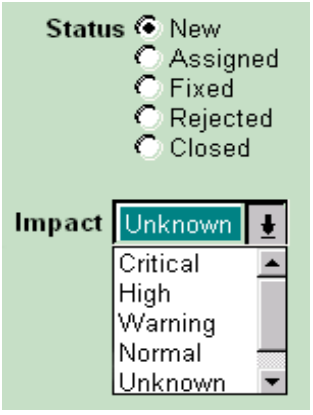
```
FIELD 1000000005 "Owned by $OWN_USER$." "Not owned."
```

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 131.

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 6-5 shows example enumerated fields in a schema.

Figure 6-5 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 message do not match the available values in the schema. To do this, you map the index numbers of the OVO 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 6-5 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 (0) to unknown (4)
- normal in OVO (1) to normal (3)

- warning in OVO (2) to warning (2)
- minor in OVO (3) to high (1)
- major in OVO (4) to high (1)
- critical in OVO (5) to critical (0)

Handling Multiple OVO Messages

NOTE This section applies to OVO for UNIX only.

You can submit multiple OVO messages to create just one action request. When you do this, the SPI for Remedy ARS has a number of values available for every variable you use in the action definition. To specify which value the SPI for Remedy ARS uses, insert a multiple message keyword before the `FIELD` keyword.

Table 6-12 Multiple Message Keywords

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

Table 6-12 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 SPI for Remedy ARS does not update any fields in the action request if the OVO message changes. To specify that the SPI for Remedy ARS should update a field when necessary, insert the `UPDATE` keyword before the `FIELD` keyword. You can extend this with more specific update keywords.

Table 6-13 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 SPI for Remedy ARS replaces the existing value only if the new value is higher. For text fields, the SPI for Remedy ARS adds the new value after the existing value. *This is the default for text fields.*
OVERWRITE	The SPI for Remedy ARS always replaces the existing value with the new value. *This is the default for numeric fields.*

Table 6-13 shows the different update keywords you can use. The example below shows examples of update policies where the SPI for Remedy ARS:

- 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 SPI for Remedy ARS 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 SPI for Remedy ARS 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 SPI for Remedy ARS 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.

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:

```
ATTACH_TO
  FIELD  9 "$MSG_OBJ$"
  FIELD  7 "Key for field 7"
SET TARGET "ProbSys"
  FIELD  9 "$MSG_OBJ$"
  UPDATE APPEND FIELD 10 "$MSG_TEXT$"
SET 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 SPI for Remedy ARS and displayed in Example 6-1 to demonstrate how the field names of an ARS schema shown in Figure 6-6 on page 137, are associated with unique identifiers, shown in Table 6-14 on page 138. This section covers the following topics:

- "Submitting OVO Messages"

- "Updating an Action Request"
- "Updating a OVO Message"

Example 6-1 illustrates how any changes to the fields you indentify may then be linked to actions that need to be carried out in OVO. The AR schema shown in Figure 6-6 enables help desk users to create and update details of reported problems.

Figure 6-6 Example Schema

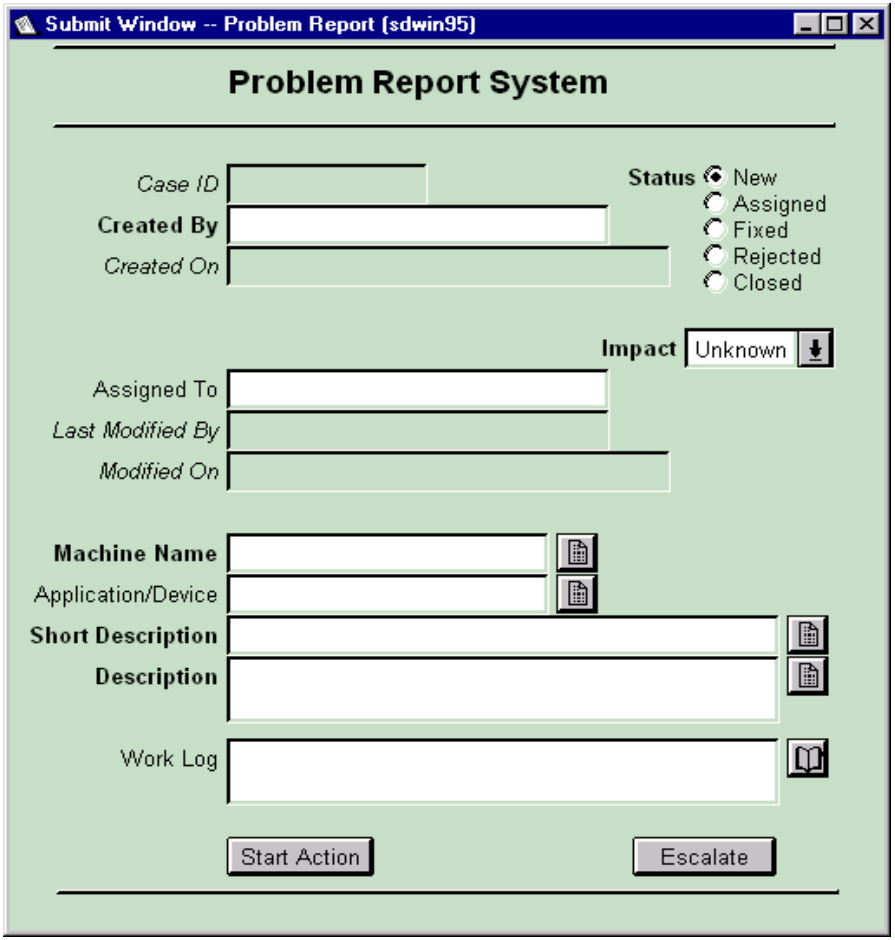


Table 6-14 shows the unique identifiers for each of the fields in the Problem Report System schema. The rules file uses the identifiers to reference the fields and their values.

Table 6-14 Example Schema Field Identifiers

Field Name	Identifier
Case ID	1
Created By	2
Created On	3
Assigned To	4
Last Modified By	5
Modified On	6
Status	7
Short Description	8
Work Log	1000000001
Impact	1000000002
Machine Name	1000000003
Application/Device	1000000004
Description	1000000005

Example 6-1 shows a simple rules file that defines how the SPI for Remedy ARS handles OVO messages and action requests in the Problem Report System.

Example 6-1 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"
```

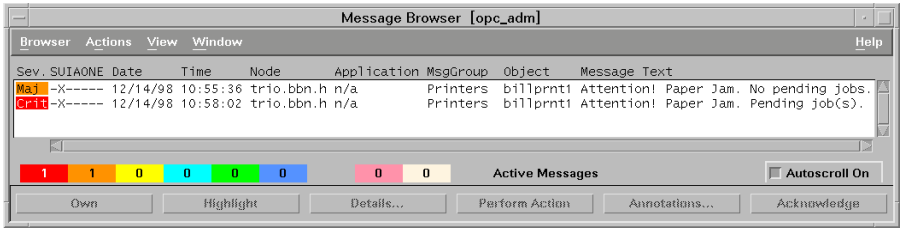
```
  CONDITION
    SEVERITY Normal

#*** 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 Messages

Figure 6-7 shows two OVO messages that appear when a large printer fails.

Figure 6-7 OVO Messages for Submission



When the OVO user decides to submit these messages to the AR System, and the following occurs:

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

4. `Condition2` in the rules file matches, so the SPI for Remedy ARS checks no further.
5. The SPI for Remedy ARS completes the default actions for the target `ProbSys`.
6. The SPI for Remedy ARS completes the actions for `Condition2`.
7. The SPI for Remedy ARS completes the actions listed in the `ON_SUBMIT`section.

Figure 6-8 Created Action Request

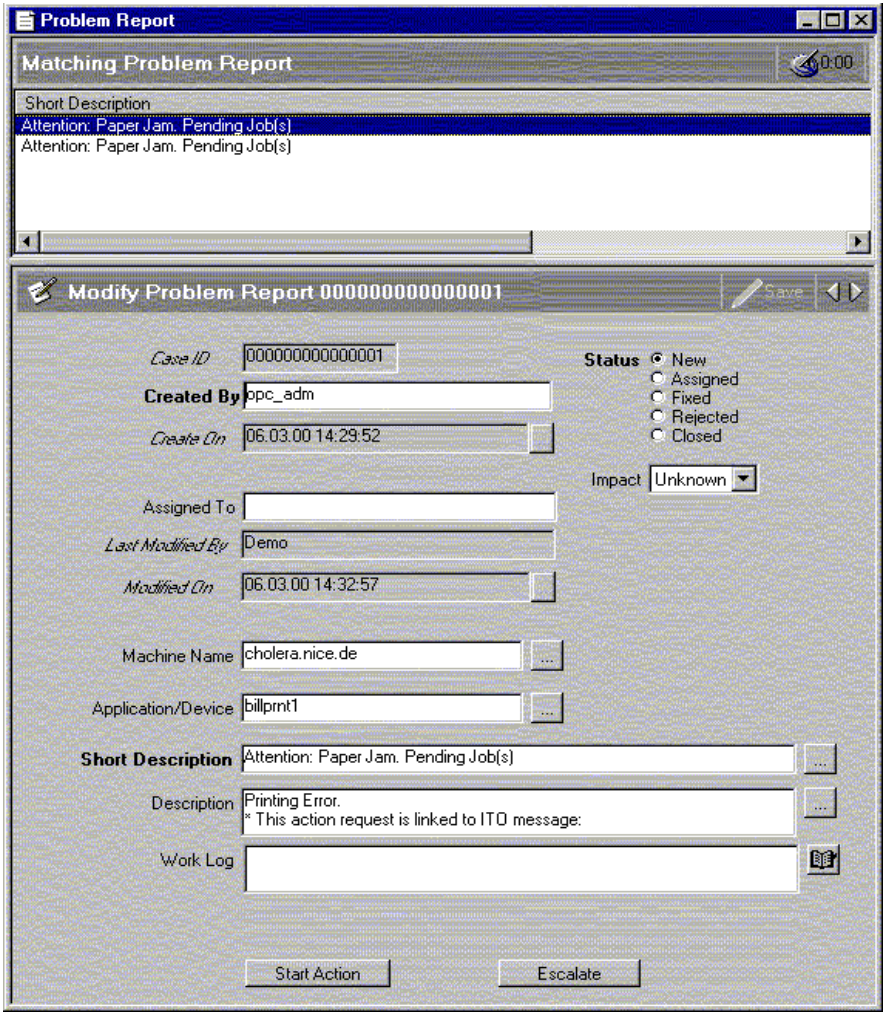


Figure 6-8 shows the action request that the SPI for Remedy ARS creates as a result of the OVO message. Notice that:

- 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 message.

NOTE

The first condition in Example 6-1 creates a different action request if the OVO message is within the `Printers` message group, but is only a warning. This involves overriding the target's default action for field `7`.

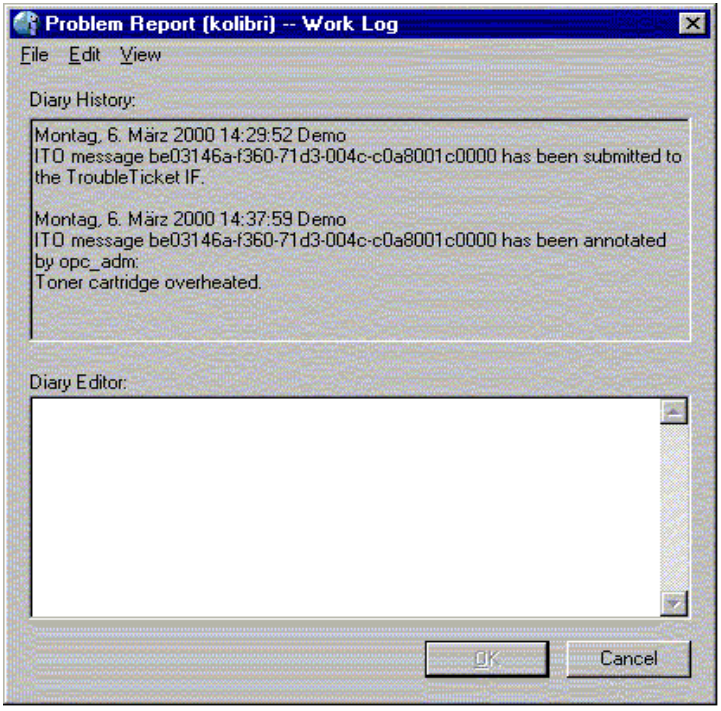
The third condition in Example 6-1 creates an action request for all critical OVO messages that do not meet the first or second conditions.

Updating an Action Request

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

Because the SPI for Remedy ARS applied the second condition in Example 6-1 when creating the action request, it also applies the second condition when updating it.

Figure 6-9 Updated Action Request



In the second condition, the SPI for Remedy ARS can update the `Work Log` field with details of a message event. Therefore, when the OVO user adds the annotation, the SPI for Remedy ARS appends details of this event to the action request's work log. Figure 6-9 shows the how the annotation in OVO has updated the work log entry in the action request.

NOTE

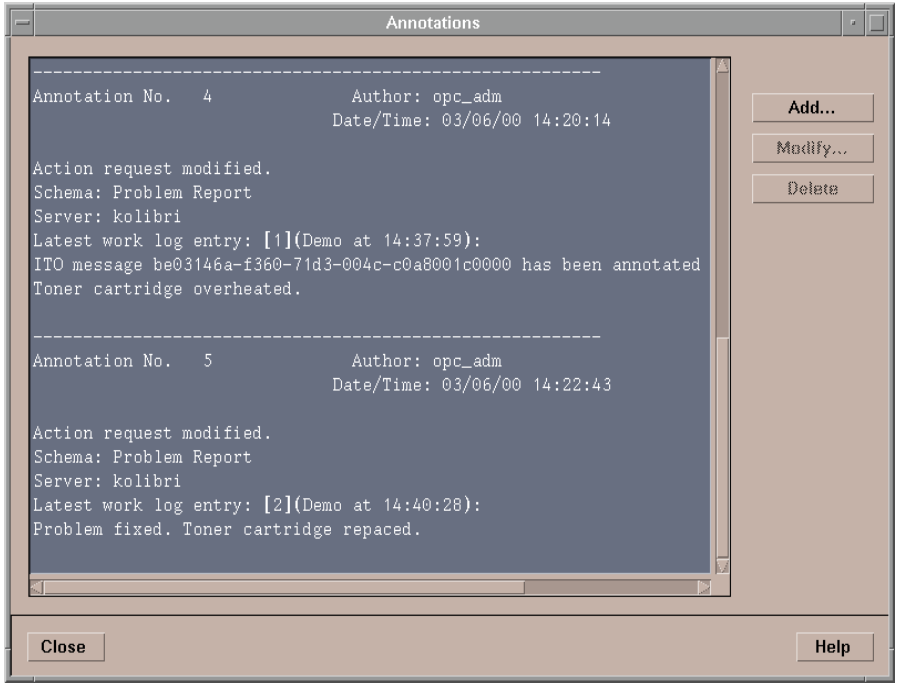
When the SPI for Remedy ARS applies the first condition in Example 6-1, it cannot update any of the action request's fields.

When the SPI for Remedy ARS applies the third condition in Example 6-1, it can only update the `Work Log` field.

Updating a OVO Message

Figure 6-10 shows how amendments to the `Status` and `Work Log` of the action request cause an annotation to be added automatically to the corresponding OVO message—the same OVO message, which created the Action Request being amended here. When the user saves the amendments, a filter calls the SPI for Remedy ARS client, which sends details of the action request and any modifications to the SPI for Remedy ARS server.

Figure 6-10 Changed Action Request



This target's update rules specify that the SPI for Remedy ARS must then:

- add an annotation to the OVO message that includes the `Work Log` entry
- take ownership of the OVO message because the `Status` is now `Assigned`

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

ARS Filters and Active Links

To enable ARS to communicate with OVO via the SPI for Remedy ARS 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 directly from the AR schema

This section describes the following topics:

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 message.

The itoupdate Command

The AR filters and active links call the SPI for Remedy ARS client. The SPI for Remedy ARS client is the `itoupdate` utility, which is copied to the AR server either during the distribution of the SPI for Remedy ARS templates or manually with the `ftp` command. For more information about template assignment and distribution, see "Installing the SPI for Remedy ARS" on page 40.

NOTE We refer to the installation directory of an OVO managed node running Windows as `IDIR`, e.g. `c:\Program Files\HP OpenView\Installed Packages\{79....}`

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 `IDIR\bin\instrumentation\itoupdat.exe`

The `itoupdate` command informs the SPI for Remedy ARS 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 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 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 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 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 netween AR servers" on page 159 for details.

The remspifilter Command

The `remspifilter` command registers the filters you create in ARS and want to use to perform actions in OVO. 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 SPI for Remedy ARS client. Enter the path to the location where you installed the SPI for Remedy ARS 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 SPI for Remedy ARS 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 153.

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:

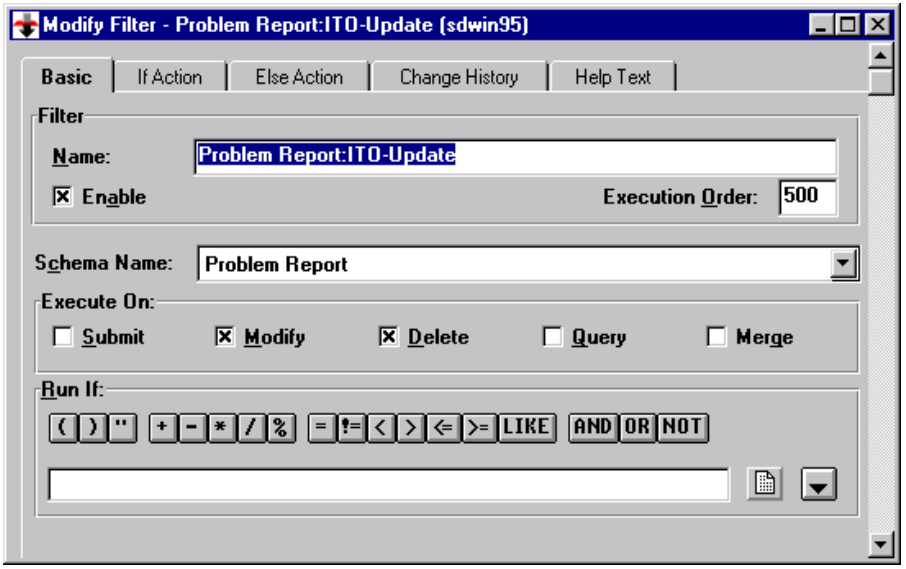
```
remspifilter -create -cmd \
/Program_Files/OpenView/Remspi/itoupdate
```

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:

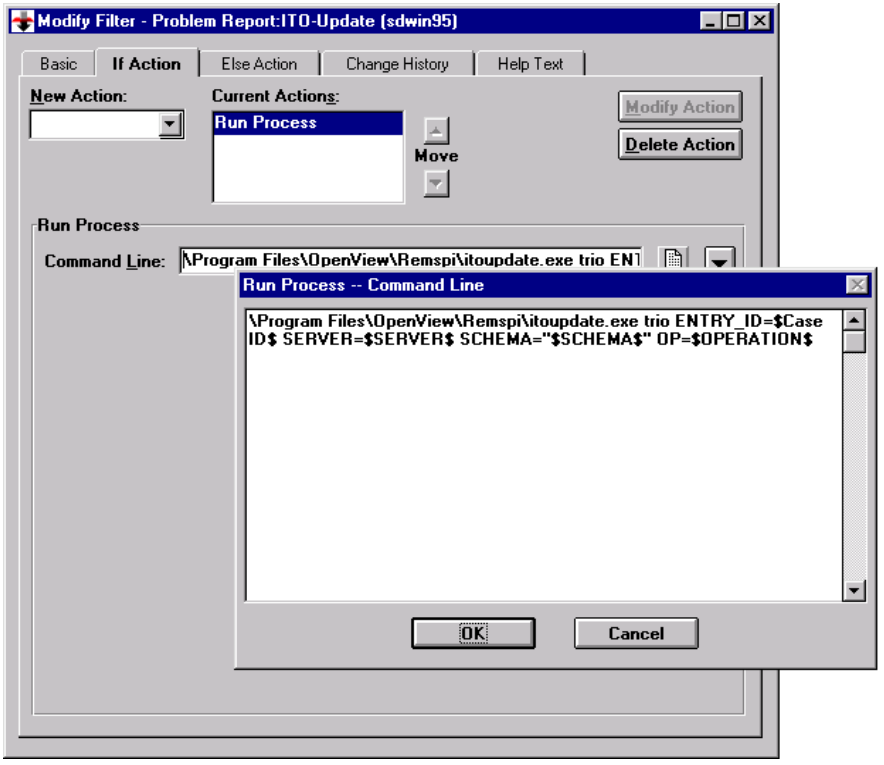
1. Create a new filter that executes when someone modifies or deletes an action request as shown in Figure 6-11.

Figure 6-11 A New Filter



2. Add a run-process action that calls the SPI for Remedy ARS client as shown in Figure 6-12.

Figure 6-12 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:

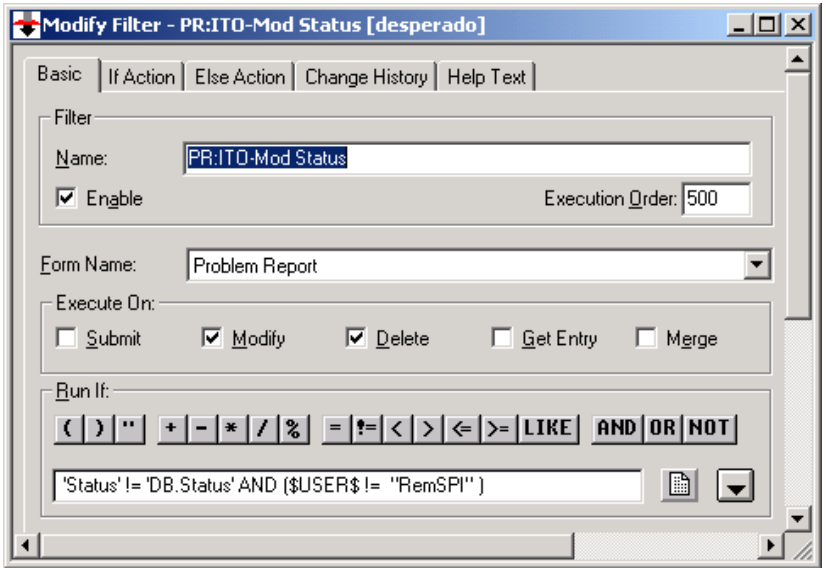
1. Create a new filter that executes when someone modifies or deletes an action request as shown in Figure 6-13.

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 6-13 The special filter for modifcations of the `Status` field



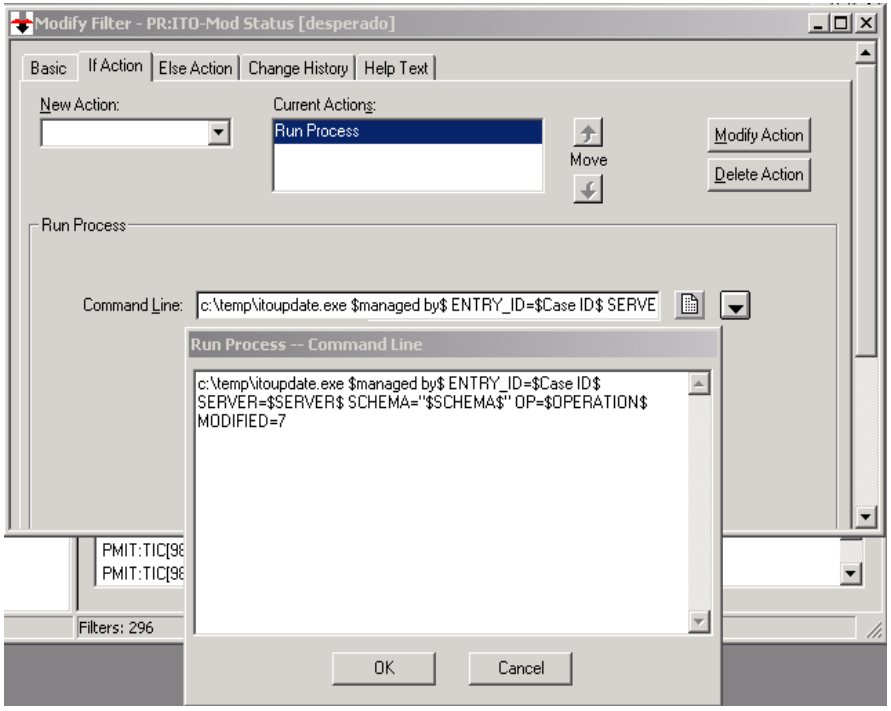
2. Add a run-process action that calls the SPI for Remedy ARS client as shown in Figure 6-14.

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 6-14 The special filter's Run Process

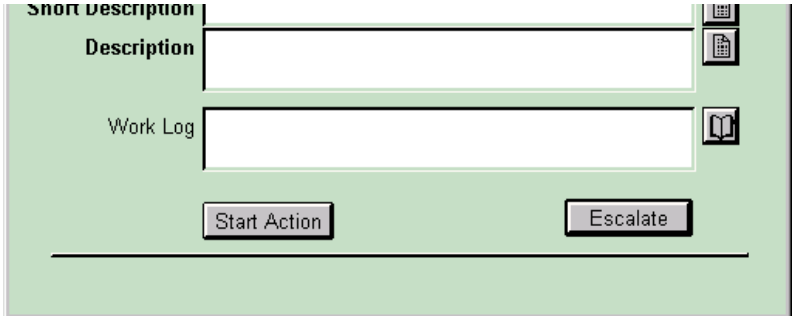


Configuring an Active Link

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

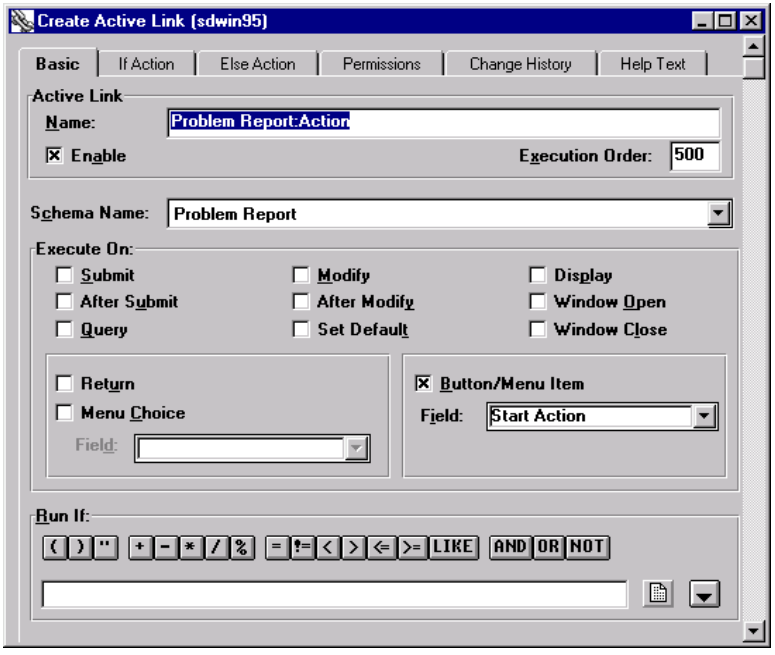
1. Use the ARS to create the appropriate buttons or menu items in the ARS schema as shown in Figure 6-15.

Figure 6-15 Buttons for Active Links



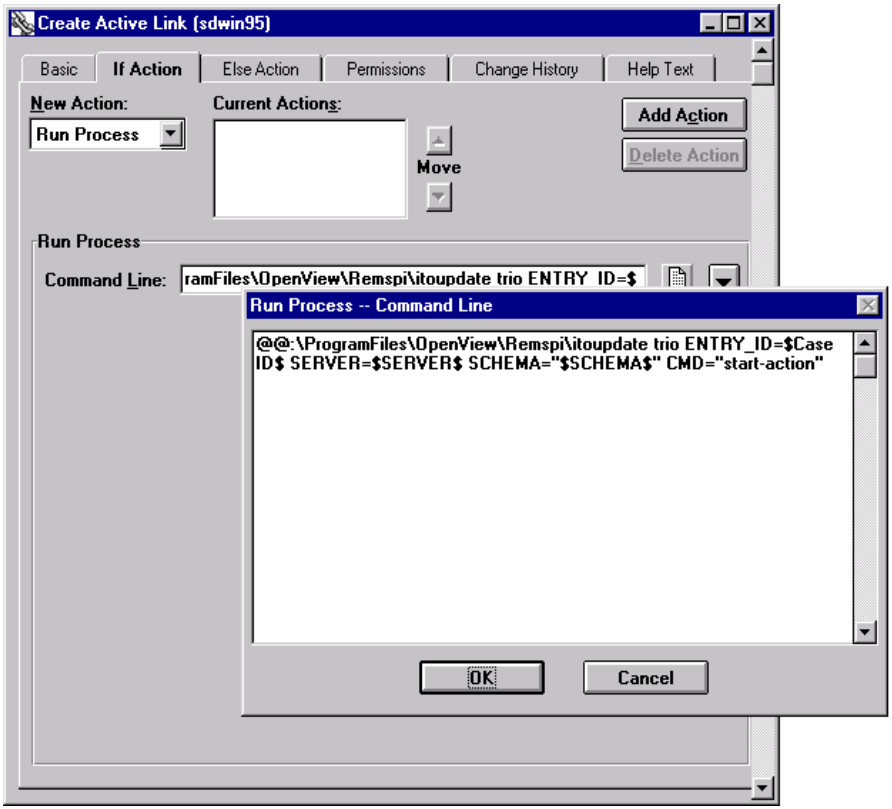
2. Create a new active link that runs when someone uses the menu item or button created in step 1.

Figure 6-16 A New Active Link



3. Add a run process that calls the SPI for Remedy ARS client as illustrated in Figure 6-17. 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 6-17 An Active Link's Run Process



Customizing the SPI for Remedy ARS for Multiple Servers

The SPI for Remedy ARS enables you to use more than one AR server, more than one OVO server, or both. If any of these situations applies for your organization, you need to customize the SPI for Remedy ARS appropriately. Figure 2-7 on page 29 shows an environment with multiple OVO and AR servers. This section provides information about the following topics:

- "Using Multiple AR Servers"
- "Using Multiple OVO Servers"

Using Multiple AR Servers

If you want the SPI for Remedy ARS 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. This section provides information about the following additional topics:

- "Moving an AR ticket netween AR servers"
- "Notify the SPI server about the ticket move"
- "Instruct the SPI server to handle the move"

Example 6-2 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 6-2 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 netween 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.

NOTE

The actual move of the AR ticket is done by the AR server and not by the SPI for Remedy ARS. 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 SPI for Remedy ARS 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 \
  <SPI for Remedy ARS 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 SPI for Remedy ARS, 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 Servers

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

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

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

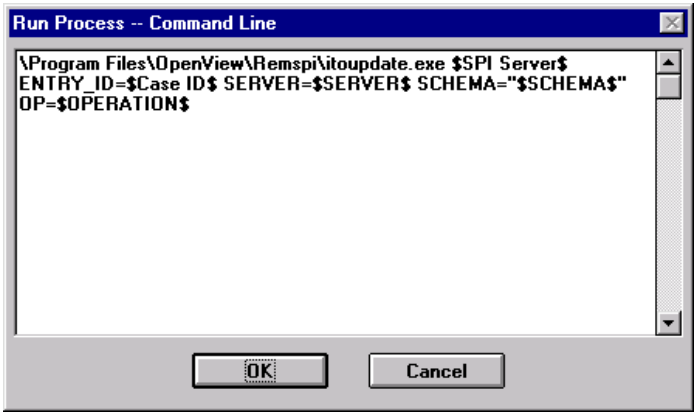
To enable this, you must complete the following steps for each schema:

1. Create a field in the schema to store the name of the SPI for Remedy ARS 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 SPI for Remedy ARS client. Reference the new field as the first parameter in the SPI for Remedy ARS client command.

   Figure 6-18 shows an example filter's run process where field `SPI Server` contains the name of the SPI for Remedy ARS server that created the action request.

The exact path and name for the SPI for Remedy ARS client program depends on the platform and your setup. For more details, see "Installing the SPI for Remedy ARS" on page 40.

Figure 6-18 Filter Run-Process for Multiple OVO Servers



3. Amend the rules file for *both* SPI for Remedy ARS servers. Each rules file must contain an action that puts the name of the SPI for Remedy ARS server into the appropriate field using the action variable `HOST_NAME`.

   The action must use the `FIRST` keyword to ensure that the SPI for Remedy ARS enters only one value. For more details about defining actions, see "Defining Actions" on page 120.

   Example 6-3 shows a default action for a target. The action puts the name of the SPI for Remedy ARS server in field `1000000006`.

Example 6-3 Default Action for Multiple OVO Servers

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

FIRST FIELD 1000000006 "$HOST_NAME$"
```

Using the SPI for Remedy ARS on cluster nodes

If the SPI for Remedy ARS is used on cluster nodes, several configuration issues have to be considered. This section provides information about the following topics:

- "AR Server installed on a cluster node"
- "OVO Server installed on a cluster node"

NOTE The following rules only describe the SPI specific requirements which have to be satisfied to use the SPI for Remedy ARS 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`.

- Comunication 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"`
- Comunication 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 SPI for Remedy ARS submits a OVO 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

To submit a OVO message to the SPI for Remedy ARS, the scripts `remspisub_auto.sh` (Trouble Ticket interface) and `remspisub_man.sh` (OVO application) pass the message ID to the `remspiupd` command, which forwards the information to the SPI for Remedy ARS 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 message does not provide. The SPI for Remedy ARS can add this extra information when it creates a new action request.

The remspiupd Command

The `remspiupd` command handles the internal communication between the SPI for Remedy ARS server and OVO. The command can be used both to submit and update OVO messages and is called by the `remspisub_*.sh` scripts. For more information on the `remspisub_*.sh` scripts, see "Data Components" on page 29.

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 message ID must be specified, and the following options are allowed:

- `-a(ttach)`

  enables one message to be *attached* to another message. Note that at least *two* OVO 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 SPI for Remedy ARS server and prints this to `stdout`. This is intended for submission from the OVO GUI to

immediately display the result of the submission. The default submission scripts use the `-q` option for automatic submission by the OVO server.

- `-u(user)`

  option allows to specifiy the OVO user submitting the message

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

- at least *one* OVO message must already have been submitted
- the already submitted OVO messages must be associated with only *one* AR
- there must be at least *one* OVO 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 173.

Using External Value Files

The SPI for Remedy ARS allows you to use external values as a source of information in a similar way to the attributes and fields of a OVO 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 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 message to the SPI for Remedy ARS 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 SPI for Remedy ARS 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 6-4 shows a simple external value file that contains extra details about a printer.

Example 6-4 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>`

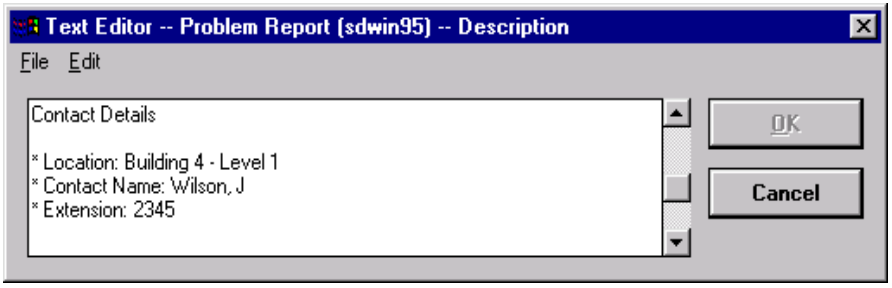
You use this like any other variable in the rules file, except that it consists of two keywords. For more details on actions, see "Defining Actions" on page 120.

Example 6-5 shows an action that puts external data from Example 6-4 into a text field. Figure 6-19 shows the result in the action request.

Example 6-5 References to External Values

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

Figure 6-19 External Values in an Action Request



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

7 Troubleshooting the SPI for Remedy ARS

This section describes how to go about troubleshooting the HP OpenView Smart Plug-In for Remedy Action Request System integration.

In this Section

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

This section describes the log files to which the SPI for Remedy ARS server writes information concerning errors and tracing. Table 7-1 on page 173 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 7-1 Logfile Locations

File Name	SPI for Remedy ARS 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 SPI for Remedy ARS 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 SPI for Remedy ARS server and information relating to the self-monitoring of the SPI for Remedy ARS server. For more information, see "Troubleshooting SPI for Remedy ARS Usage" on page 181.

The `remspiupd.trace` contains trace information relating to attempts by the SPI for Remedy ARS to submit OVO messages to the ARS and attach modifications to existing OVO messages. For more information, see "The remspiupd Command" on page 166. Note that this file is not written by default.

The `remspi_submit.log` contains information relating to the manual or automatic submission of OVO messages to the ARS. For more information, see "Troubleshooting SPI for Remedy ARS Usage" on page 181.

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 SPI for Remedy ARS configuration file `remspi.cfg`.

The keywords used for tracing are listed in Table 7-2 on page 174.

Table 7-2 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 on UNIX management servers using the command sequence
  `ovstop RemedySPI; ovstart RemedySPI`

- Restart the SPI server on Windows management servers using the command sequence
  `net stop 'HP OV SPI for Remedy Service' & net start 'HP OV SPI for Remedy Service'`

As an alternative on UNIX management servers, 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 on UNIX management server s using the command sequence
  `ovstop RemedySPI; ovstart RemedySPI`
- Restart the SPI server on Windows management servers using the command sequence
  `net stop 'HP OV SPI for Remedy Service' & net start 'HP OV SPI for Remedy Service'`

As an alternative on UNIX management servers, 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 SPI for Remedy ARS server process manually using the following command.

On UNIX management servers. enter:

```
/opt/OV/bin/remspi/remspisrv -debug
```

On Windows management servers. enter:

```
ProgramFiles\HP OpenView\bin\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 SPI for Remedy ARS Installation

To verify that the installation of the HP OpenView Smart Plug-In for Remedy Action Request System integration completed successfully on the OVO management server, carry out the steps in the following list:

1. On the command line of an OVO for UNIX management server, enter: **`swlist`**

   Look for the following entries:

   ```
   SPI-Remedy B.01.10
   ```

   ```
   SPI-RemedyDoc B.01.10
   ```

2. In the OVO 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 adm
	`remspi-op` / RemSPI op
Policy Groups	`SPI for Remedy ARS - AR Server NT`
	`SPI for Remedy ARS - AR Server UX`
	`SPI for Remedy ARS - ITO`

3. If either of these steps fails to produce the desired results:

   a. On OVO for UNIX management servers,

check the following log files for more specific information relating to problems concerning the installation of the SPI for Remedy ARS:

- `/var/adm/sw/swagent.log`
- `/var/adm/sw/swinstall.log`

b. On OVO for Windows management servers,

check the following log files for more specific information relating to problems concerning the installation of the SPI for Remedy ARS:

- *`ProgramFiles`*`\HP OpenView\Data\HPOVInstall\RemedySPI_install.log`

c. Cross check your installation with the information provided in Table 3-9, “File Locations on the OVO Management Server,” on page 47.

NOTE

After you install the SPI for Remedy ARS 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 SPI for Remedy ARS server, you will need to configure an alternative in the `/etc/services` file on the SPI for Remedy ARS server machine - the OVO fpr UNIX management server; and in `%WINDIR%\system32\drivers\etc\services` on the OVO fpr Windows management server.

To modify port a number:

1. As user `root` on the OVO management server, open the `services` file in a text editor.
2. Find the section that begins with the comment - or add a new section if it’s not already in the file:

   ```
   # HP OpenView OVO 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 SPI for Remedy ARS Rules

The `rules` file determines how the SPI for Remedy ARS should react to messages from OVO that are intended for ARS, and action requests from the AR system, which are intended for OVO. For more information about the syntax and structure of the `rules` file, see "Customizing the Rules File" on page 95.

NOTE The SPI for Remedy ARS server will not start if the `rules` file contains any errors. To check for errors, use the `remspiconfig` command, which is located in the directory containing the SPI for Remedy ARS binaries.

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 7-1 shows the messages that appear when the rules file references a non-existent schema field.

Example 7-1 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' ...
Conenction succeeded.

All done.
```

The lines beginning with `Can't validate...` enable you to locate the error in the rules file. In Example 7-1, 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 SPI for Remedy ARS. 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 SPI for Remedy ARS 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:

SPI for Remedy ARS Monitors

- Symptom

  If you acknowledge (or take ownership of) a large number of OVO messages that are linked to ARS, the overall performance of the SPI for Remedy ARS can be significantly affected.

- Explanation

  The SPI for Remedy ARS 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 request queue, `RemSPI_QSize`, on the SPI for Remedy ARS server. `RemSPI_QSize` holds SPI for Remedy ARS-related OVO messages with a severity level of either `warning` or `critical`. If the number of messages in `RemSPI_QSize` exceeds 1000, the SPI for Remedy ARS 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 SPI for Remedy ARS. However, the SPI for Remedy ARS 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 SPI for Remedy ARS target definitions, see "Defining Targets" on page 99.

Responsibility of the REMSPI_ITO_USER (OVO for UNIX only)

- 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 teh 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 enviroment, 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 SPI for Remedy ARS 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 153 for details on how to configure.

8 De-installing the SPI for Remedy ARS

This section describes how to de-install the SPI for Remedy ARS software from the systems on which you have installed it.

In this Section

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

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 151.

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 SPI for Remedy ARS from the OVO GUI

The SPI for Remedy ARS integration with the OVO GUI has to be removed manually. You will have to remove the following components from the OVO GUI:

- Managed Nodes Groups

  Note that you should not remove the AR servers from the managed nodes repository until you have removed the OVO agent software including the SPI for Remedy ARS policies. For more information, see "Removing SPI for Remedy ARS Software from the AR Server" on page 190.

- Message Groups (OVO for UNIX only)
- Tools and Tool Groups
- Policies and Policy Groups
- Users (OVO for UNIX) or User Roles (OVO for Windows)

For more information on how to remove elements from the OVO GUI, see the *HP OpenView Operations Administrator's Reference*.

Removing the SPI for Remedy ARS from the OVO Management Server

To remove the HP OpenView Smart Plug-In for Remedy Action Request System integration components from the OVO management server and complete the general clean up process:

1. On the OVO for UNIX management server, use the `swremove` command. Enter:

   `swremove SPI-Remedy`

2. On the OVO for Windows management server, use the Windows `Add/Remove Software` Icon in the Windows `Control Panel`, select the entry for the `HP OpenView Smart Plug-ins` and press the `Change` button. Then answer the questions and finally mark the `Remedy SPI` check box before starting the removal process.

3. Remove the OVO management server and the AR servers from the SPI for Remedy ARS specific node groups. Then re-deploy the policies using the normal OVO means to these nodes.

4. Cleanup the GUI by removing the SPI for Remedy ARS-specific elements. For more information, see “Removing the SPI for Remedy ARS from the OVO GUI” on page 188.

Removing SPI for Remedy ARS Software from the AR Server

After you have removed from the OVO GUI those elements added by the SPI for Remedy ARS and removed the SPI for Remedy ARS software from the OVO management server, you need to remove the SPI for Remedy ARS software from the AR server.

To remove the SPI for Remedy ARS instrumentation from the AR server, re-deploy the instrumentation to the AR server using OVO means. On OVO for Windows, select the check box `Remove existing instrumentation ...` before starting the deployment.

After you have removed the SPI for Remedy ARS installation from the AR servers, you can remove the AR servers from the managed nodes repository in the OVO GUI.

Index

Index

S

W

Index

Index

Index

Index

Index

Index

Index

Index

Index

Index

Index

Index

Index

Index