

# **HP Operations Manager for Windows Smart Plug-in for Remedy Action Request System**

## **Administrator's Reference**

**Software Version 4.06**

**For Microsoft Windows® operating systems**

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



**Manufacturing Part Number: None**

**Document Release Date: August 2013**

**Software Release Date: August 2013**

© Copyright 2013 Hewlett-Packard Development Company, L.P.

---

## Legal Notices

### Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notices

©Copyright 1998-2013 Hewlett-Packard Development Company, L.P., all rights reserved.

### Trademark Notices

Acrobat®, Adobe®, and PostScript® are trademarks of Adobe Systems Incorporated.

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

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

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

For information about third-party license agreements, see the `spi-thirdparty-license-agreements` directory in the product installation directory.

---

# Contents

<b>1. SPI for Remedy ARS Online Help</b>	
In this Section. . . . .	12
<b>2. Introducing the SPI for Remedy ARS</b>	
In this Section. . . . .	14
Overview of the SPI for Remedy ARS . . . . .	15
Understanding HP Operations Manager for Windows . . . . .	16
Messages. . . . .	16
Message Management . . . . .	17
Actions . . . . .	17
Understanding the Action-Request System . . . . .	19
The ARS Work Flow . . . . .	19
The Action-Request Form . . . . .	20
The Functionality of the SPI for Remedy ARS . . . . .	23
Key Features. . . . .	23
Example Scenarios . . . . .	24
The SPI for Remedy ARS Architecture. . . . .	27
Process and Network Components. . . . .	27
Data Components . . . . .	29
SPI for Remedy ARS Components Integrated with HPOM . . . . .	32
<b>3. Installing the SPI for Remedy ARS</b>	
In this Section. . . . .	34
Installation Prerequisites . . . . .	35
Disk-Space Requirements . . . . .	37
Memory (RAM) Requirements . . . . .	37
Installing the SPI for Remedy ARS 4.06. . . . .	38
Installation Overview . . . . .	38

---

# Contents

Installing the SPI for Remedy ARS Software on HPOM for Windows . . . . .	39
Verifying Installation of the SPI for Remedy ARS Software . . . . .	40
Assigning the SPI for Remedy ARS Policies . . . . .	41
Deploying SPI for Remedy ARS Policies in HPOM for Windows . . . . .	41
Installed File Locations . . . . .	42
Operating the SPI for Remedy ARS in a “non-root agent” environment . . . . .	43
<b>4. Getting Started</b>	
In this Section . . . . .	46
Configuration Overview . . . . .	47
Uploading the Example Form to ARS . . . . .	48
Configuring Communication Between HPOM and ARS . . . . .	50
Configure the SPI for Remedy ARS in HPOM . . . . .	50
Verifying the Configuration . . . . .	52
Configuring Communication between ARS and HPOM . . . . .	53
Configuring the itoupdate Command . . . . .	54
Registering the Remedy-ARS Filters . . . . .	55
Communicating with the Action-Request Server through a Firewall . . . . .	56
Using Private Queues on the Action-Request Server . . . . .	56
Verifying the Configuration . . . . .	57
<b>5. Using the SPI for Remedy ARS</b>	
In this Section . . . . .	60
SPI for Remedy ARS Node Groups . . . . .	61
SPI for Remedy ARS Users . . . . .	62
SPI for Remedy ARS Tool Groups . . . . .	63
SPI for Remedy ARS Administrator Tools . . . . .	64
The SPI for Remedy ARS Server’s Status . . . . .	64

---

# Contents

Starting and Stopping the SPI for Remedy ARS Server . . . . .	64
Testing the SPI for Remedy ARS . . . . .	65
Starting and Stopping the Action-Request Server . . . . .	65
Manually Submitting Messages to an Action-Request System . . . . .	66
<b>6. Customizing the SPI for Remedy ARS</b>	
In this Section. . . . .	68
Customizing HP Operations Manager for Windows . . . . .	69
Customizing HPOM GUI Elements . . . . .	70
Modifying the SPI for Remedy ARS Policies . . . . .	71
Setting up the Trouble-Ticket Interface on HPOM . . . . .	73
Customizing the Rules File . . . . .	76
The Structure of the Rules File. . . . .	76
Defining Global Options . . . . .	78
Defining Targets . . . . .	82
Defining Buffer Rules . . . . .	85
Defining Update Rules. . . . .	86
Defining Submit and Attach Rules. . . . .	97
Defining Move-From and Move-To Rules. . . . .	103
Defining Conditions . . . . .	104
Defining Actions . . . . .	111
An Example of a Rules File . . . . .	127
ARS Filters and Active Links . . . . .	137
Method 1: The itoupdate command . . . . .	138
Method 2: The ItoUpdatePlugin plugin . . . . .	140
The remspifilter command . . . . .	141
Configuring the ARS Filter automatically . . . . .	143
Configuring the ARS Filter manually for itoupdate. . . . .	143
Configuring the IF_MODIFIED ARS Filter manually for itoupdate. . . . .	145
Configuring an Active Link for itoupdate . . . . .	147

---

# Contents

Configuring the ARS Filter manually for ItoUpdatePlugin . . . . .	149
Customizing the SPI for Remedy ARS for Multiple Servers . . . . .	154
Using Multiple Action-Request Servers . . . . .	154
Using Multiple HPOM Servers . . . . .	157
Using Backup Action-Request-System Servers . . . . .	159
Maintaining a backup copy of the SPI database . . . . .	160
Using the SPI for Remedy ARS in Cluster and Load Balancer Environments . . . . .	162
Remedy ARS Server on a Cluster Node . . . . .	162
HPOM Server on a Cluster Node . . . . .	163
Two Remedy Servers behind a Load Balancer . . . . .	163
Using External Actions and Data . . . . .	165
Starting External Actions . . . . .	165
The remspiupd Command . . . . .	165
Using External Value Files . . . . .	167
Formatting External Value Files . . . . .	168
Referencing External Values . . . . .	169
Miscellaneous Configurations Tasks . . . . .	170
Setting the Date and Time Format . . . . .	170
Utilities installed with the SPI . . . . .	171
Direct access to AR tickets: remspi_arutil . . . . .	171
<b>7. Troubleshooting the SPI for Remedy ARS</b>	
In this Section . . . . .	174
Error Logs and Tracing . . . . .	175
Log File Locations . . . . .	175
Traditional Tracing . . . . .	175
Enhanced Tracing . . . . .	178
Tracing the Rules . . . . .	180

---

# Contents

Troubleshooting the SPI for Remedy ARS Installation.....	181
Troubleshooting SPI for Remedy ARS Rules .....	183
Troubleshooting SPI for Remedy ARS Usage.....	185
Name Resolution .....	185
Removing Buffered Action Requests.....	185
Setting the Language of the SPI server .....	186
Duplicated annotations.....	187
Missing annotations for ITO_UPDATE actions.....	187
Remedy AR server installed in non standard directory .....	188
Troubleshooting the SPI for Remedy ARS Internal Database.....	189
Troubleshooting the SPI for Remedy ARS Performance .....	191
Handling Message Storms .....	193
<b>8. De-installing the SPI for Remedy ARS</b>	
In this Section.....	196
Removing Action-Request Filters .....	197
Removing the SPI for Remedy ARS from the HPOM GUI .....	198
Removing the SPI for Remedy ARS from the HPOM Server.....	199
Removing the SPI for Remedy ARS from the Remedy ARS Server .....	200

---

# Contents



---

## Support

Please visit the HP Operations Manager for Windows web site at:

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

This web site provides contact information and details about the products, services, and support that HP Operations Manager for Windows offers.

HP Operations Manager for Windows 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 enhancement requests online
- Download software patches
- 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](http://support.openview.hp.com/access_level.jsp)

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

[https://ovrd.external.hp.com/rd/register;jsessionId=C98D8B79ACC32E8714D436513D14C35D?target=HTTPS%3a%2f%2fovrd.external.hp.com%2fhpp%2fhpp2rd%3fdest\\_url%3dhttp%3a%2f%2fsupport.openview.hp.com%2f%26hpappid%3dovrd](https://ovrd.external.hp.com/rd/register;jsessionId=C98D8B79ACC32E8714D436513D14C35D?target=HTTPS%3a%2f%2fovrd.external.hp.com%2fhpp%2fhpp2rd%3fdest_url%3dhttp%3a%2f%2fsupport.openview.hp.com%2f%26hpappid%3dovrd)



---

# **1** **SPI for Remedy ARS Online Help**

This section describes what you can find in the online help for the Smart Plug-in for Remedy Action Request System.

## **In this Section**

The on-line help for the Smart Plug-in for Remedy Action Request System 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 software and helps you to understand the basic concepts and functionality of both HPOM and Remedy ARS.
- [“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 SPI for Remedy ARS.
- [“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 SPI 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 SPI for Remedy ARS.
- [“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 provides an overview of the Smart Plug-in for Remedy Action Request System software.

## **In this Section**

This section introduces the HP Operations Manager for Windows Smart Plug-in for Remedy Action Request System software. The section covers the following areas:

- “Overview of the SPI for Remedy ARS”
- “Understanding HP Operations Manager for Windows”
- “Understanding the Action-Request System”
- “The Functionality of the SPI for Remedy ARS”
- “The SPI for Remedy ARS Architecture”

## Overview of the SPI for Remedy ARS

The Smart Plug-in for Remedy Action Request System (SPI for Remedy ARS) enables you to integrate HP Operations Manager for Windows (HPOM) with the Remedy Action Request System (ARS).

Figure 2-1

### Integration between HPOM and an Action-Request System

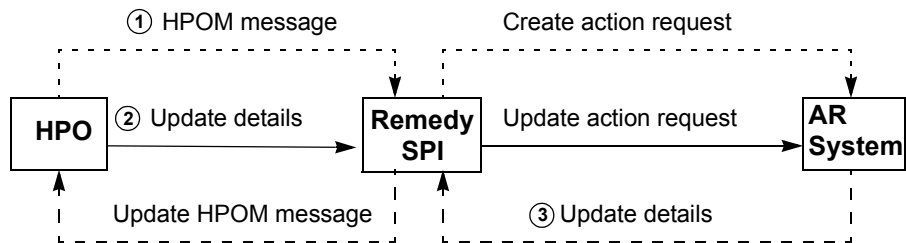


Figure 2-1 gives an overview of how the SPI for Remedy ARS integrates HPOM and an action-request system. It shows that:

1. The SPI for Remedy ARS accepts messages from HPOM, then forwards the HPOM message to the action-request system. This creates a new action request which corresponds to the HPOM message.
2. If an HPOM 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 HPOM message details.

For example, if HPOM receives a message about a printer failure, the SPI for Remedy ARS uses this message to create a new action request in the action-request 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 HPOM 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 HPOM.

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

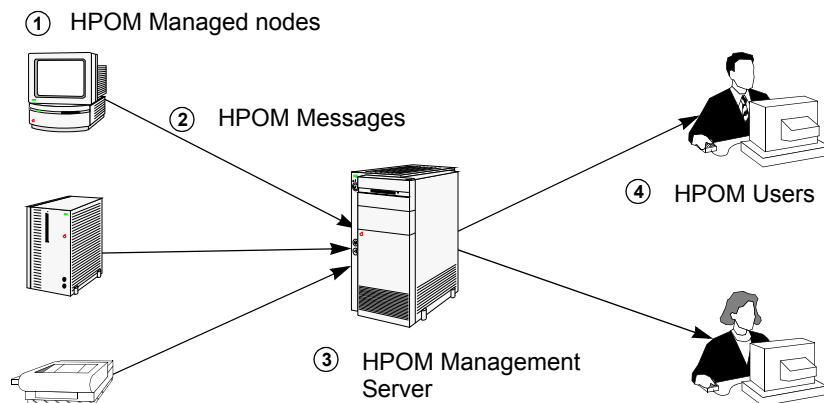
---

## Understanding HP Operations Manager for Windows

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

- “Messages”
- “Message Management”
- “Actions”

Figure 2-2 Overview of HPOM



### Messages

Figure 2-2 gives an overview of HPOM. It shows the following:

1. HPOM can monitor and control remote systems, known as managed nodes.
2. If a particular status or event occurs on a managed node, the HPOM 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 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 HPOM 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 receive a message, HPOM enables them to work on it in the following different ways:

- *Take ownership* of a message:  
Taking ownership of a message prevents other users working on the message and shows that the new owner is aware of the message and intends to deal with it personally.
- *Add annotations* to a message:  
The user can attach notes to a message; the notes provide helpful 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 to the history database. This usually happens when work on this message is complete.

## Actions

Figure 2-2 shows how messages enable HPOM 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 Action-Request System

This section gives a brief overview of the main features of the Remedy Action-Request System (ARS) that are important in the context of the SPI for Remedy ARS. The sections describe the following areas:

- “The ARS Work Flow”
- “The Action-Request 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 HPOM 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 actionrequest 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 action request 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 action request 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.

After an action request has been created, it resides in the action-request system’s 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 Action-Request Form

The action-request 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:

- “Action-Request Filters” on page 21
- “AR Active Links” on page 21

Figure 2-3 shows a simple action-request 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 Action-Request Schema

The screenshot shows a window titled "Submit Window -- Problem Report (sdwin95)". The main heading is "Problem Report System". The form contains the following fields and controls:

- Case ID**: Text input field.
- Created By**: Text input field.
- Created On**: Text input field.
- Status**: Radio button group with options:  New,  Assigned,  Fixed,  Rejected,  Closed.
- Impact**: Dropdown menu currently showing "Unknown".
- Assigned To**: Text input field.
- Last Modified By**: Text input field.
- Modified On**: Text input field.
- Machine Name**: Text input field with a help icon.
- Application/Device**: Text input field with a help icon.
- Short Description**: Text input field with a help icon.
- Description**: Text input field with a help icon.
- Work Log**: Text input field with a help icon.

The action-request 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 action-request 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.

### Action-Request Filters

A **filter** is defined in ARS as any action or group of actions carried out by ARS on the action-request 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 action-request filter mechanism to trigger communication between the ARS and HP Operations Manager for Windows. The SPI for Remedy ARS client, which resides on the action-request server is registered as a filter and configured to react to events such as *modify* and *delete*. In this way, changes to action-request forms that are of interest to HPOM can trigger automatic updates in the HPOM 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 137](#).

### AR Active Links

An **active link** is defined in ARS as any action or group of actions carried out by ARS on the action-request 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.

## Introducing the SPI for Remedy ARS

### **Understanding the Action-Request System**

The SPI for Remedy ARS uses this feature to provide buttons in action-request forms which allow the user to launch operator-initiated actions in HPOM from within ARS. For more information on active links, see [“ARS Filters and Active Links”](#) on page 137.

## The Functionality of the SPI for Remedy ARS

The SPI for Remedy ARS enables you to integrate HPOM's monitoring and control capabilities with Remedy's action-request system. When HPOM 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 HPOM 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 provides the following features:

- Adapt to fit an organization's individual needs
- Create new action requests by automatically sending a HPOM message to an action-request system
- Create new action requests when an HPOM user decides to send one or more HPOM messages to an action-request system
- Use external data and actions to add supplementary details to an action request
- Update action requests when the corresponding HPOM message changes
- Update action requests by adding further HPOM messages
- Update HPOM messages from the action-request system. This includes:
  - change ownership of the HPOM message
  - add annotations to the HPOM message
  - acknowledge an HPOM message to reflect the status of the action request
  - start an action in HPOM

The SPI for Remedy ARS also provides the following functionality:

- Complete integration with the HPOM GUI
- The capacity to monitor action-request system availability with HPOM
- The capacity to monitor SPI for Remedy ARS availability with HPOM

## Example Scenarios

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

Figure 2-4

### An Action Request that Provides Information

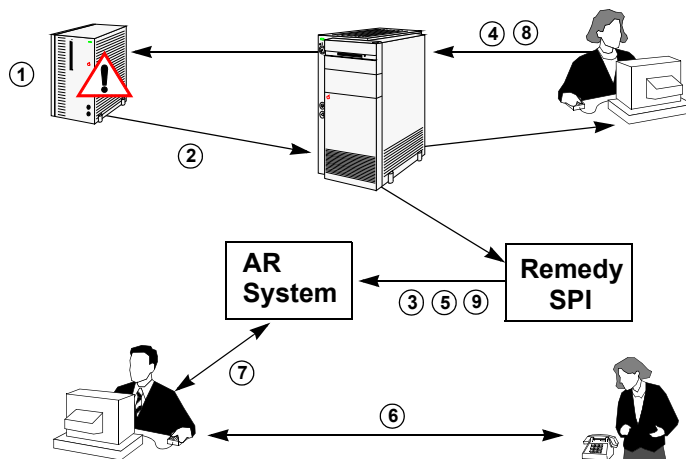


Figure 2-4 shows an example of how the SPI for Remedy ARS can create and update an action request to provide information.

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

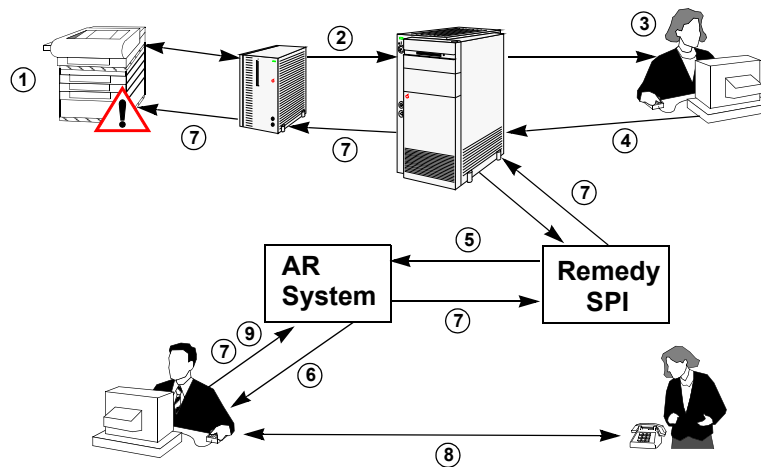


Figure 2-5 shows an example of how the SPI for Remedy ARS can create an action request that needs attention, then update the HPOM message when the action request changes.

1. A large printer in the print room fails because of a paper jam.
2. The HPOM agent on the managed node sends a message to the HPOM management server at a different site.
3. The HPOM operator sees the message and diagnoses the problem. The HPOM operator adds an annotation to the HPOM message explaining that someone needs to remove the jammed paper and then reset the printer.
4. Since the HPOM operator cannot remove the jammed paper from a printer in a remote site, he decides to forward the HPOM message to the action-request system.

**The Functionality of the SPI for Remedy ARS**

5. The SPI for Remedy ARS receives the HPOM 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 HPOM operator `Remedy SPI User` automatically takes ownership of the HPOM message linked to the Action Request. Ownership of an HPOM message indicates to other HPOM operators that the problem is being solved.

7. The help desk staff start an action in HPOM 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 HPOM 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 HPOM”

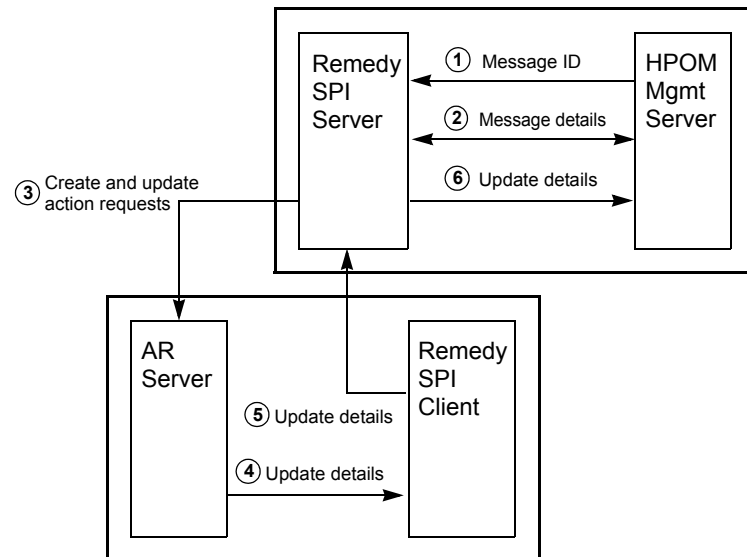
### Process and Network Components

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

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

Figure 2-6

### Server and Network Architecture



## Introducing the SPI for Remedy ARS

### The SPI for Remedy ARS Architecture

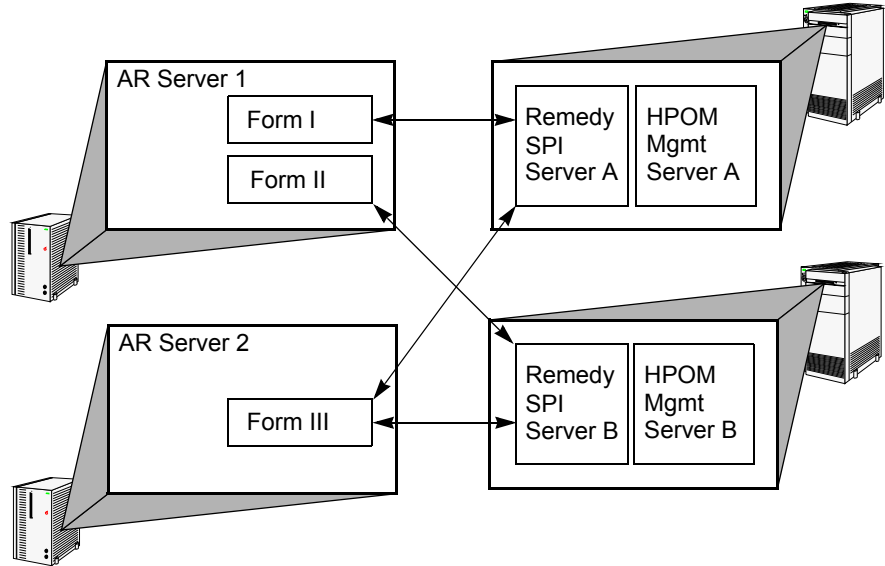
Figure 2-6 on page 27 shows that the SPI for Remedy ARS server runs on the same machine as the HPOM management server. The action-request server can run on the same or a different machine. The SPI for Remedy ARS client must be on the same machine as the action-request server. Figure 2-6 also shows:

1. The SPI for Remedy ARS server receives the ID of an HPOM message from the HPOM management server via a submission program.
2. The SPI for Remedy ARS server retrieves details of the HPOM message from the HPOM 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 action-request 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 HPOM message on the HPOM management server using HPOM's API.
7. When an HPOM message is modified, the SPI for Remedy ARS server receives a message-change event via the HPOM API.
8. HPOM sends the details of the message-change event to the SPI for Remedy ARS server via the HPOM API.

9. The SPI for Remedy ARS server updates the action request by means of the Remedy-ARS application-programming interface (API).

Figure 2-7

### Multiple Servers



The SPI for Remedy ARS also supports multiple HPOM management servers and multiple action-request 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 action-request servers. In addition, each action-request server has a SPI for Remedy ARS client that can update HPOM messages on any of the HPOM management servers.

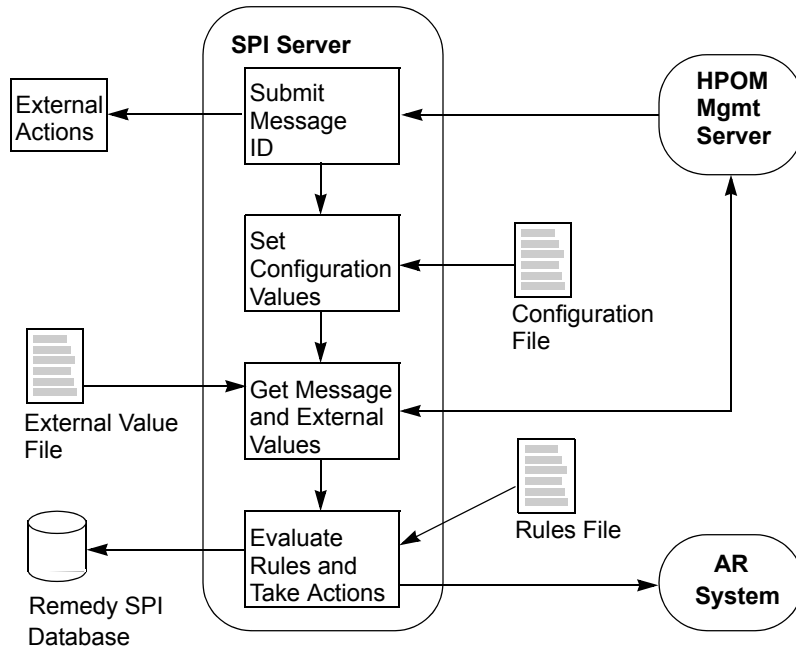
### Data Components

This section describes what data components are enabled and used when data flow between HP Operations Manager 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 an HPOM message is submitted to the action-request system.

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



### External Actions and Data

To submit an HPOM message to the SPI for Remedy ARS, HPOM passes the message's ID to a script called `remspisub_auto.vbs` (via a special WMI policy). This script has 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 HPOM 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 HPOM. In HPOM for Windows the user names and passwords are actually not used, but nevertheless need to be specified. The passwords are encrypted. For more details about creating and modifying the configuration file, see [“Configuring Communication Between HPOM and ARS” on page 50](#).

## Rules File

The rules file uses HPOM message attributes and additional external values to determine answers to the following questions:

- On which action-request system should an action request be created?
- Which action-request form (schema) should be used?
- Which HPOM message attributes match which action-request form fields?

The `rules` file defines *targets* and *conditions*. A target defines the *name* of a form on an action-request server. The target also defines how the SPI for Remedy ARS should update an HPOM message if one of this schema’s action requests changes.

The rules file also has a set of conditions. Conditions define criteria that filter HPOM messages and determine what actions need to be taken to address the problems described in the message. If an HPOM 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 HPOM 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 76](#).

## The SPI for Remedy ARS Database

The SPI for Remedy ARS maintains information in the databases relating to the associations between action requests and HPOM messages. Every time the SPI for Remedy ARS creates a new action request, it adds a new record containing the HPOM message ID and the action request ID. There is one database for each target in the rules file, because one HPOM message may have associations with several targets.

When a HPOM 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 HPOM**

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

- Node groups for:
  - AR Servers UX
  - AR Servers AIX
  - AR Servers NT
  - Remedy SPI Servers
- Default policies to monitor:
  - AR server processes
  - AR System log files
  - Remedy SPI server processes
  - Remedy SPI log files
- User roles
- Applications or tools to:
  - 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 action-request server
  - stop an action-request server



---

## **3** **Installing the SPI for Remedy ARS**

This section describes how to install the Smart Plug-in for Remedy Action Request System.

## **In this Section**

This section describes how to install the Smart Plug-in for Remedy Action Request System 4.06. The section covers the following areas:

- [“Installation Prerequisites” on page 35](#)
- [“Installing the SPI for Remedy ARS 4.06” on page 38](#)

---

### **NOTE**

The SPI for Remedy ARS 4.06 is available as a patch release.

The patch that is appropriate for your HPOM management server needs to be installed after installing the base version 4.0x of the SPI for Remedy ARS.

The 4.0x version that is appropriate for your HPOM management server is available on the HP Operations Manager Smart Plug-Ins DVD.

---

## Installation Prerequisites

The SPI for Remedy ARS consists of client and server components and integrates with existing HPOM and action-request servers. The SPI for Remedy ARS client runs on the same machine as the action-request server. The SPI for Remedy ARS server runs on the same machine as the HPOM server. This section also provides information about the following topics:

- “Disk-Space Requirements” on page 37

Table 3-1 shows the software prerequisites for the HPOM for Windows servers.

Table 3-1

**Supported HPOM for Windows Servers**

Product Version	Operating System	
	Windows	
	2003 Server	2008 Servers
HPOM for Windows 9.0	X	X

### NOTE

The SPI for Remedy ARS is linked with Remedy ARS libraries 7.1.0 (API version 13). For a client program using the 7.1.0 libraries, the list of possible target action-request server versions includes 7.5 to 8.1, but other restrictions may apply. Please consult the support matrix for the Remedy ARS server for details.

Table 3-2 shows software prerequisites for the action-request servers on the various supported hardware platforms.

**Table 3-2 Prerequisites for the Remedy Action-Request Server**

Hardware Platform	Operating System	AR System Version
		7.5, 7.6, 8.0 <sup>a</sup> , 8.1 <sup>b</sup>
HP	HP-UX 11.11, 11.23, 11.31 <sup>c</sup>	X

**Table 3-2 Prerequisites for the Remedy Action-Request Server**

Hardware Platform	Operating System	AR System Version
		7.5, 7.6, 8.0 <sup>a</sup> , 8.1 <sup>b</sup>
Intel	Windows Server 2003, 2008, 2012 <sup>d</sup>	X
Sun Sparc	Solaris 9, 10, 11	X
IBM RS6000	AIX 5.3 - 7.1 <sup>e</sup>	X
Intel	SuSE SLES 10, 11 <sup>f</sup>	X
	RedHat RHEL 4.2 - 6.3 <sup>gh</sup>	X

- a. If `remspifilter` is used to create or delete AR filters, the AR form must not be of type *Customized*, but needs to be of type *Unmodified* or *Overlay*.
- b. If `remspifilter` is used to create or delete AR filters, the AR form must not be of type *Customized*, but needs to be of type *Unmodified* or *Overlay*.
- c. PA-RISC and IA64.
- d. 32/64 Bit
- e. 32/64 Bit
- f. 32/64 Bit
- g. 32/64 Bit
- h. IPv6 must not be disabled, although it doesn't need to be configured. If it is disabled, the SPI client will not be able to resolve names and IP addresses. In order to enable, add the following two lines to `/etc/modprobe.conf` and reboot:
 

```
alias ipv6 on
options ipv6 disable=0
```

---

**NOTE**

To take full advantage of the SPI for Remedy ARS, your action-request servers should be HPOM managed nodes. The prerequisites listed above do not invalidate the prerequisites for the action-request server mentioned by the manufacturer (BMC/Remedy Corp.).

---

## Disk-Space Requirements

Table 3-3 lists the disk space requirements for the SPI for Remedy ARS software on both the HPOM management server and the action-request server, which is the SPI for Remedy ARS client.

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

Server	HPOM Version	OS	Install (MB)	Runtime Files (MB)	Total (MB)
HPOM Server	all	UNIX	200	10 + X <sup>a</sup>	200 + X
	all	Windows	15	5 + Y <sup>b</sup>	15 + Y
ARS Server <sup>c</sup>	n/a	n/a	n/a	5	5

- a. Approx. 100 bytes per message forwarded to the ARS server and not yet acknowledged in HPOM. If the global option `FORGET_ACKED_MSGS` is not turned on, this applies for each message forwarded to (but not yet deleted on) the ARS server.
- b. Approx. 50 bytes per message (in one file) forwarded to the ARS server and not yet acknowledged in HPOM. If the global option `FORGET_ACKED_MSGS` is not turned on, this applies for each message which forwarded to (but not yet deleted on) the ARS server.
- c. SPI for Remedy ARS client

## Memory (RAM) Requirements

There are no specific requirements concerning the amount of RAM installed on either the HPOM or the action-request server.

## Installing the SPI for Remedy ARS 4.06

The installation steps differ very slightly depending on whether or not you want your action-request server to be an HPOM managed node. The instructions in this section assume that you *do* want your action-request 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 action-request server processes. If you do *not* want your action-request servers to be HPOM managed nodes, please pay special attention to the steps described in [“Installation Overview” on page 38](#).

The information in this section covers the following topics:

- [“Installation Overview” on page 38](#)
- [“Installing the SPI for Remedy ARS Software on HPOM for Windows” on page 39](#)
- [“Verifying Installation of the SPI for Remedy ARS Software” on page 40](#)
- [“Assigning the SPI for Remedy ARS Policies” on page 41](#)
- [“Deploying SPI for Remedy ARS Policies in HPOM for Windows” on page 41](#)
- [“Installed File Locations” on page 42](#)

### Installation Overview

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 HPOM management server
3. Verify that the installation of SPI for Remedy ARS software bundle completed successfully on the HPOM management server
4. Assign and distribute the appropriate SPI for Remedy ARS policies to the HPOM management server
5. Either:
  - a. Assign and distribute the appropriate SPI for Remedy ARS policies, along with all the rest of the SPI for Remedy ARS software, to the action-request server (the HPOM managed node)

or:

- b. If you do *not* want your action-request server to be an HPOM managed node, copy the `itoupdate` program (`itoupdate.exe` for Windows) file from the HPOM management server to the action-request server manually.

Ideally, the file should be copied to the location indicated in [Table 3-6, “File Locations for the Installed SPI for Remedy ARS Client,”](#) on page 42. 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 HP Operations Manager for Windows. For more information, see [“ARS Filters and Active Links”](#) on page 137.

## **Installing the SPI for Remedy ARS Software on HPOM for Windows**

The Smart Plug-in for Remedy Action Request System 4.06 is available as a patch release, which needs to be installed *after* installing the base version 4.0x. The base version 4.0x is available on the appropriate *HP Operations Manager Smart Plug-Ins DVD*.

- For a *HPOM 8 management server*, first install the base version 4.00, which is available on the *HP Operations Manager Smart Plug-Ins DVD 2009*. Then install the patch `REMSPIOMW8_00001`.
- For a *HPOM 9 management server*, first install the base version 4.03, which is available on the *HP Operations Manager Smart Plug-Ins DVD 2010.3*. Then install the patch `REMSPIOMW9_00001`.

To install the Smart Plug-in for Remedy Action Request System software bundle on the HPOM management server, carry out the following steps:

1. On the HPOM management server, insert the appropriate HP Operations Manager Smart Plug-Ins DVD into the DVD drive.
2. Follow the instructions of the installation wizard and select the Smart Plug-in for Remedy Action Request System to install. This will install the base version.
3. Install the patch that is appropriate for your management server and base version (see above). The installation instructions are contained in the patch documentation.

4. Copy the following Action Request System server libraries into the directory %OvInstallDir%\bin\REMSPI. The ARS server libraries may be found on your original Action Request System Install CDs.

- arapi71.dll
- arrpc71.dll
- arut171.dll

These commands install the SPI software and perform basic configuration. The software bundle contains all the SPI for Remedy ARS software, configuration files, and documentation.

### **Verifying Installation of the SPI for Remedy ARS Software**

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

1. In the HPOM GUI, check that the following elements have been added:

- Node Groups:
  - Remedy SPI Servers
  - AR Servers NT
  - AR Servers UX
  - AR Servers AIX
- Application/Tool Groups:
  - Remedy SPI Admin
- User Roles:
  - Remspi\_adm
  - Remspi\_op
- Policy Groups:
  - Remedy SPI - AR Server NT
  - Remedy SPI - AR Server UX
  - Remedy SPI - AR Server AIX
  - Remedy SPI - ITO



## Assigning the SPI for Remedy ARS Policies

To facilitate the assignment of the SPI for Remedy ARS policies to the various HPOM and action-request servers in your environment, you first need to add the servers to HPOM 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 HPOM management server. For details on how to configure HPOM please consult the products manual.

1. Start the HPOM GUI.
2. Add the node for the action-request servers - if they are not already defined in HPOM.
3. Copy the action-request servers to the appropriate SPI node group, according to their platforms.
4. Copy the HPOM 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 3-4](#). Consequently, any managed node that you copy to the node groups listed has the correct policies assigned automatically.

Table 3-4

**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
AR Servers AIX	SPI for Remedy ARS - AR Server AIX
Remedy SPI Servers	SPI for Remedy ARS - ITO

## Deploying SPI for Remedy ARS Policies in HPOM for Windows

To distribute the SPI for Remedy ARS policies to the HPOM for Windows management server and, assuming it is to be an HPOM managed node, the action-request server, too, no additional steps are necessary.

The installation of the SPI for Remedy ARS marks the necessary policies as auto deployable so that the HPOM for Windows management server automatically distributes the policies to the nodes that you add to the SPI for Remedy ARS node groups.

## Installed File Locations

The installation process copies the necessary files to the HPOM management server. [Table 3-5](#) shows the directories that the installation process creates on the HPOM management server for the SPI for Remedy ARS.

**Table 3-5 File Locations on the HPOM Management Server**

File Type	Directory
Binaries	%OvInstallDir%\bin\remspi
Libraries	%OvInstallDir%\bin\remspi
Temporary / runtime	%OvShareDir%\tmp\remspi
Log files	%OvDataDir%\log\remspi
Installation Log files	%OvShareDir%\SPI_Install\remspi
Configuration	%OvShareDir%\conf\remspi
HPOM integration	%OvInstallDir%\install\remspi
Documentation	%OvInstallDir%\NLS\1033\Help
Message Catalogs	%OvShareDir%\conf\remspi\remspi.cat

[Table 3-6](#) shows the directories for SPI for Remedy ARS clients that you install on a HPOM managed node. These are the standard locations for HPOM commands.

**Table 3-6 File Locations for the Installed SPI for Remedy ARS Client**

Operating System	Directory
UNIX	/var/opt/OV/bin/instrumentation
Windows	%OvAgentDir%\bin\instrumentation

---

## Operating the SPI for Remedy ARS in a “non-root agent” environment

The SPI for Remedy ARS is able to run on a managed node where the HPOM agent is started under a non-root account (Unix agents only). See the according HPOM documentation for more information about how to configure the HPOM agent for such an environment.

After you switch the HPOM agent to support a non root environment, you have to configure the SPI for Remedy ARS, too, as follows:

1. Switch the HPOM agent to non root
2. Execute the following script on the managed node as user “root”:

```
/var/opt/OV/bin/instrumentation/remspi_root.sh
```

It is essential that you perform the last step *after* the HPOM agent has been switched to non root. This is because the HPOM step changes permissions and ownerships of the deployed instrumentation. The SPI for Remedy ARS needs this information to perform its own switch.

Installing the SPI for Remedy ARS

**Operating the SPI for Remedy ARS in a “non-root agent” environment**

---

# 4 **Getting Started**

This section describes how to configure the Smart Plug-in for Remedy Action Request System.

## **In this Section**

This section describes how to install and upgrade the HP Operations Manager for Windows Smart Plug-in for Remedy Action Request System. The information in this section covers the following areas:

- [“Configuration Overview”](#)
- [“Uploading the Example Form to ARS”](#)
- [“Configuring Communication Between HPOM and ARS”](#)
- [“Configuring Communication between ARS and HPOM”](#)

## Configuration Overview

The information in this section describes a simple scenario that takes you through the steps required to set up communication between HP Operations Manager for Windows and the Remedy Action Request System by means of the SPI for Remedy ARS. The instructions in this section assume that you have already successfully installed the SPI for Remedy ARS software on the HPOM management server and assigned and distributed the SPI for Remedy ARS templates to both the HPOM management server and the action-request server. For more information about the installation of the software and the assignment and distribution of the SPI for Remedy ARS templates, see [“Installing the SPI for Remedy ARS 4.06” on page 38](#).

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

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

1. Configure the HPOM management server to use the SPI for Remedy ARS server to communicate with your ARS server(s).
2. Configure the ARS server(s) to use the SPI for Remedy ARS client to communicate with the HPOM management server.

To help you understand how the configuration process works and allow you to verify whether or not the configuration has completed successfully, the SPI for Remedy ARS provides a very simple example action-request form, which you can upload to the action-request server and quickly configure. For ease of use, the same form appears elsewhere in the book, for example; in [“Customizing the SPI for Remedy ARS” on page 67](#)

## Uploading the Example Form to ARS

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

To upload the example form to ARS:

1. Depending on the operating system installed on the action-request server, use the `ftp` command to transfer the appropriate file from the HPOM management server to the action-request server. You can find the files on the management server in the following location:

```
%OvShareDir%\conf\remspi
```

The file to use depends on the platform and type of the HPO agent:

- *DCE* agent:
  - Windows: `ProblemReport-NT.def`
  - AIX: `ProblemReport-AIX.def`
  - Other UNIX: `ProblemReport-UX.def`
- *HTTPS* agent:
  - Windows: `ProblemReport-NT-HTTPS.def`
  - UNIX: `ProblemReport-UX-HTTPS.def`

2. In the ARS administrator's GUI on the action-request server, select the action-request 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, e.g.:

```
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 HPOM and ARS

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

- [“Configure the SPI for Remedy ARS in HPOM” on page 50](#)  
Configure the SPI for Remedy ARS so that it knows to which action-request servers to send the messages it receives from HPOM, what conditions apply, and what schemas exist in ARS
- [“Verifying the Configuration” on page 52](#)  
Test the connection you have set up between HPOM and ARS

For more information on adding and modifying users in HPOM, see [“SPI for Remedy ARS Users” on page 62](#). For information on how to set up the Trouble-Ticket (TT) interface in HPOM so that HPOM messages can be automatically submitted to the ARS via the SPI for Remedy ARS, see [“Setting up the Trouble-Ticket Interface on HPOM” on page HIDDEN](#).

### Configure the SPI for Remedy ARS in HPOM

In this section, you tell the SPI for Remedy ARS about the action-request 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 HPOM via the SPI for Remedy ARS. You define these details in the `rules` file, which you have to create in the following place on the HPOM management server:

```
%OvShareDir%\conf\remspi
```

In this directory, you'll find a template named `rules.PR` which you may use as a starting point.

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 76](#).

---

**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 HPOM:

1. Open the `rules` file with a text editor
2. Enter or modify the following values in the TARGET section:
 

SERVER	Name of the action-request server where <i>SCHEMA</i> is defined. This value is <i>mandatory</i> : no default value is possible.
SCHEMA	Name of the form (schema) to be modified by the HPOM message
USER	User name to log in to the action-request server where <i>SCHEMA</i> is defined. This is the account the SPI for Remedy ARS server uses to log in to the action-request server. Note that you should use different account names for the SPI for Remedy ARS server and for interactive action-request users to distinguish SPI for Remedy ARS server triggered transactions from normal action-request user transactions.
PASSWORD	Password of <i>USER</i>
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 HPOM message generated by the condition match
4. Save the modifications to the `rules` file.
5. On the HPOM management server, restart the SPI for Remedy ARS server process to enable the new configuration.

Enter (or use the Windows Control Panel to restart the service):

```
net stop RemSpiSrv
net start RemSpiSrv
```

---

**NOTE**

If an action-request server which is configured in the `rules` file is not available during the startup of the SPI for Remedy ARS, a warning is written to the SPIs log file, but the SPI is nevertheless started.

The verification of the AR-related configuration, which is necessary for proper operation of the SPI, is retried later at regular intervals.

As soon as an action-request server is reachable, the verification is performed and an entry is written to the SPI for Remedy ARS's log file.

---

### Verifying the Configuration

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

1. In the HPOM GUI, start the tool `Remedy SPI Admin:SPI Test`
2. Verify that the HPOM message you send creates an Action Request in the Remedy-ARS GUI:
  - a. Start the Remedy-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 HPOM from the list at the top of the window
  - e. Verify the values in the action-request schema that is displayed at the bottom of the window

---

## Configuring Communication between ARS and HPOM

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

- [“Configuring the itoupdate Command” on page 54](#)

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

- [“Registering the Remedy-ARS Filters” on page 55](#)

Register Remedy action-request filters so that they can use the `itoupdate` command to update HPOM messages.

- [“Communicating with the Action-Request Server through a Firewall” on page 56](#)

Configure the SPI for Remedy ARS server to use a special, fixed port to communicate with the action-request server.

- [“Using Private Queues on the Action-Request Server” on page 56](#)

Define a private queue for the SPI server communication with the action-request server.

- [“Verifying the Configuration” on page 57](#)

Verify that you have successfully configured the `itoupdate` command to update HPOM messages automatically as a result of changes to the status of action requests.

Note that the `itoupdate` command is installed on the action-request server either when you install the SPI for Remedy ARS software on the action-request server (the HPOM managed node) or when you copy the `itoupdate` file from the HPOM management server to the action-request server manually.

## Configuring the itoupdate Command

This step describes how to use the `rules` file on the HPOM management server to configure the way in which the `itoupdate` command reacts to ARS filters on the action-request 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 other than the details concerning login and server. 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 HPOM management server:

```
%OvShareDir%\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`
- `ANNOTATE`
- `CMA`
- `SEVERITY`
- `MSGTXT`

[“An Example of a Rules File” on page 127](#) 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 48](#). For more detailed information about linking changes to Action Request fields with changes to the status of HPOM messages, see [“Defining Update Rules” on page 86](#).

## Registering the Remedy-ARS Filters

This step describes how to register action-request-system filters on the HPOM management server so that they can use the `itoupdate` command to update HPOM 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 141](#)

To register a Remedy-ARS filter:

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

Enter (or use the Windows Control Panel to stop the service):

```
net stop RemSpiSrv
```

2. Use the command `remspifilter` to register the Remedy-ARS filters on the HPOM management server.

```
%OvInstallDir%\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 82](#).

3. Confirm the activation of the filter. Enter:

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

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

Enter (or use the Windows Control Panel to start the service):

```
net start RemSpiSrv
```

---

### NOTE

Note that the automatic generation of action-request filters is only available for filters not using the `IF_MODIFIED` tag. Filters using the `IF_MODIFIED` tag have to be created manually using the Remedy-ARS administrator’s GUI.

---

## Communicating with the Action-Request Server through a Firewall

If there is a firewall between the action-request server and the HPOM management server or the communication port of the action-request server has been restricted in some other way, the SPI for Remedy ARS server can use a special, fixed port to communicate with the action-request server.

In the file `%OvShareDir%\conf\remspi\rules`, locate the TARGET section for the action-request server and insert a line “ARTCPPORT portnum”, where `<portnum>` is the number of the port the action-request server listens for requests.

A global default for this communication port may be defined in the file `%OvShareDir%\conf\remspi\remspi.cfg`. If there is no global default and no specific definition in the rules file, the communication port is set to 0 and thus the portmapper is used to determine the actual port number. The global default may be defined by inserting a line “REMSPI\_ARTCPPORT portnum” in the file `remspi.cfg`.

---

### NOTE

Note that this situation requires configuration on the action-request server, too. For details, see the *Remedy Action Request System Administrator's Reference*.

---

## Using Private Queues on the Action-Request Server

If a private queue should be used for the SPI server communicating with the action-request server, you can specify the number of this `rpc` queue.

In the file `%OvShareDir%\conf\remspi\rules`, locate the TARGET section for the action-request server and insert a line “ARRPC rpcnum”, where `rpcnum` is the number of the `rpc` port used by the action-request server.

A global default for this private queue number may be defined in the file `%OvShareDir%\conf\remspi\remspi.cfg`. This default may be defined by inserting the line “REMSPI\_ARRPC rpcnum” in the file `remspi.cfg`.

---

### NOTE

Note that this situation requires configuration on the action-request 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 HPOM messages automatically as a result of changes to the status of action requests, you can use the Remedy Action Request System (ARS) to modify the action request you created when you send a HPOM message in “Verifying the Configuration” on page 52. To verify that the addition or update of comments to an Action Request are reflected in the annotations attached to the original HPOM message:

1. Submit a HPOM message to ARS as described in “Verifying the Configuration” on page 52
2. Login to the action-request server and open the Problem Report Schema
3. In ARS, select the action request created by the HPOM 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 HPOM, 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 requests in ARS are automatically reflected in a corresponding change of status to the appropriate messages in HPOM:

1. In ARS, select the action request created by the HPOM message you have just submitted and open the action request for modification
2. Change the status of the action request to `Assigned`
3. In HPOM, 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 HPOM, 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.

Getting Started

**Configuring Communication between ARS and HPOM**

---

## **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 provides 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”](#)
- [“SPI for Remedy ARS Users”](#)
- [“SPI for Remedy ARS Tool Groups”](#)
- [“SPI for Remedy ARS Administrator Tools”](#)
- [“Manually Submitting Messages to an Action-Request System”](#)

## SPI for Remedy ARS Node Groups

New node groups are installed with the Smart Plug-in for Remedy Action Request System on an HPOM management server. These node groups are:

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

You should add the action-request servers in your ARS environment to the AR Servers NT, AR Servers UX and AR Servers AIX node group as appropriate.

The system running the SPI for Remedy ARS server, normally the HPOM 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 3-4 on page 41](#). 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 Remedy ARS versions that are supported by the Smart Plug-in for Remedy Action Request System, see [“Installation Prerequisites” on page 35](#).

## **SPI for Remedy ARS Users**

This section describes the default SPI for Remedy ARS user roles that appear in the `User Roles` window after you have installed the SPI. The items are:

- `Remspi_adm`
- `Remspi_op`

The default HPOM user roles for the SPI for Remedy ARS, `remspi_adm` and `remspi_op` are intended as examples: they can be used as a basis for the creation of “real” HPOM user accounts, which are added by the HPOM administrator. The default SPI for Remedy ARS user accounts have access to the default SPI for Remedy ARS applications to allow them to manage the SPI for Remedy ARS (starting and stopping action-request 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.

## **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 Smart Plug-in for Remedy Action Request System creates two new tool groups specifically for the SPI for Remedy ARS. The new tool groups are:

- **Remedy SPI Admin**  
contains tools for the administration of the SPI for Remedy ARS

---

### **NOTE**

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

---

For more information about the contents of the **Remedy SPI Admin** tool group, see [“SPI for Remedy ARS Administrator Tools” on page 64](#).

## **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 HPOM. The `Remedy SPI Admin` tool group by default contains the following tools:

- [“The SPI for Remedy ARS Server’s Status” on page 64](#)
- [“Starting and Stopping the SPI for Remedy ARS Server” on page 64](#)
- [“Testing the SPI for Remedy ARS” on page 65](#)
- [“Starting and Stopping the Action-Request Server” on page 65](#)

---

### **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 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.

### **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 76](#).

---



## Testing the SPI for Remedy ARS

As an administrator you can test the status of the SPI for Remedy ARS using the SPI Test tool . A message appears in the HPOM Message Browser window indicating the result of the test.

## Starting and Stopping the Action-Request Server

For maintenance purposes, you can start or stop an action-request server that runs on an HPOM managed node. To do this:

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

## **Manually Submitting Messages to an Action-Request System**

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

To manually submit HPOM messages to a Remedy Action-Request System:

1. In the `Message Browser` window, select one message that you want to submit.
2. Open the message text details for this message.
3. Change the message text and add the text `ToARS :` at the beginning of the text. Note that this special prefix may be changed in the appropriate WMI policy which is part of the SPI.

The message submission to ARS fails if one of the following conditions applies:

- The HPOM message does not match any of the conditions in the rules file. For more details, see [“Defining Conditions” on page 104](#).
- HPOM or an HPOM user has already submitted the same HPOM message. You cannot resubmit HPOM 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 86](#).

If message submission is successful, a new action request appears in the appropriate action-request schema.

---

## **6** **Customizing the SPI for Remedy ARS**

This section describes how to customize the SPI for Remedy ARS to suit the demands of your environment.

## **In this Section**

This section describes how to customize the Smart Plug-in for Remedy Action Request System to suit the demands of your environment and 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 HP Operations Manager for Windows”
- “Customizing the Rules File”
- “ARS Filters and Active Links”
- “Customizing the SPI for Remedy ARS for Multiple Servers”
- “Using Backup Action-Request-System Servers”
- “Maintaining a backup copy of the SPI database”
- “Using the SPI for Remedy ARS in Cluster and Load Balancer Environments”
- “Using External Actions and Data”
- “Miscellaneous Configurations Tasks”
- “Utilities installed with the SPI”

## **Customizing HP Operations Manager for Windows**

The SPI for Remedy ARS's installation process creates a default configuration within HPOM. 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 HPOM GUI Elements” on page 70](#)
- [“Modifying the SPI for Remedy ARS Policies” on page 71](#)
- [“Setting up the Trouble-Ticket Interface on HPOM” on page 73](#)

### Customizing HPOM GUI Elements

After you install the SPI for Remedy ARS, new elements appear in the HP Operations Manager for Windows 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 HPOM 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 action-request server runs. By default, this window is empty to allow you to add nodes as appropriate.
	AR Servers UX	UNIX (non AIX) nodes on which an action-request server runs. By default, this window is empty to allow you to add nodes as appropriate.
	AR Servers AIX	AIX nodes on which an action-request server runs. By default, this window is empty to allow you to add nodes as appropriate.
Application/Tool Group	Remedy SPI Admin	Applications that the SPI for Remedy ARS administrator uses to manage the SPI for Remedy ARS and action-request servers.

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

Object Type	Name	Description
User Roles	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	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 in the HPOM GUI. For example, you might want to modify the configuration of a policy to change the polling interval or set up the Trouble-Ticket interface for individual message conditions.

The following tables describe the SPI for Remedy ARS policies.

**Table 6-2** Policy Group “Remedy SPI - AR Server AIX”

Policy Name	Description
RemSPI ARS Server errors (UX)	Monitors the AR system server error log file (UNIX version)
RemSPI itoupdate messages (AIX)	Monitors the log file of itoupdate (AIX version)
RemSPI_AR_server_UX	Check whether Remedy ARS processes are down (UNIX version)

**Table 6-2 Policy Group “Remedy SPI - AR Server AIX” (Continued)**

Policy Name	Description
RemSPI_ARS_server_UX_Up	Check whether Remedy ARS processes are up (UNIX version)

**Table 6-3 Policy Group “Remedy SPI - AR Server NT”**

Policy Name	Description
RemSPI ARS Server errors (NT)	Monitors the AR system server error log file (Windows version)
RemSPI itoupdate messages (NT)	Monitors the log file of itoupdate (Windows version)
RemSPI_ARS_server_NT	Check whether Remedy ARS processes are down (Windows version)
RemSPI_ARS_server_NT_Up	Check whether Remedy ARS processes are up (Windows version)

**Table 6-4 Policy Group “Remedy SPI - AR Server UX”**

Policy Name	Description
RemSPI ARS Server errors (UX)	Monitors the AR system server error log file (UNIX version)
RemSPI itoupdate messages (UX)	Monitors the log file of itoupdate (UNIX version)
RemSPI_ARS_server_UX	Check Remedy ARS processes

**Table 6-5 Policy Group “Remedy SPI - ITO”**

Policy Name	Description
RemSPI SPI Server	Monitors the error log file of the SPI server process
RemSPI Internal Messages	Catches internal messages of the SPI



**Table 6-5 Policy Group “Remedy SPI - ITO” (Continued)**

Policy Name	Description
RemSPI Test	Catches test messages of the SPI
RemSPI Submit TT - Manual	Catches messages which are to be manually submitted by modifying their message text to start with “ToARS:”.
RemSPI Submit TT	Automatically submit new messages which have the “Forward to TT” flag set.
RemSPI Update TT - Severity	Forward a message change event to the SPI (change of message severity).
RemSPI Update TT - Text	Forward a message change event to the SPI (change of message text).
RemSPI Update TT - State	Forward a message change event to the SPI (change of message state).
RemSPI Monitor SPI Server	Monitors the existence of the SPI server.

### Setting up the Trouble-Ticket Interface on HPOM

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

The Windows Management Interface type policy `Remedy SPI - ITO / SubmitTTtoRemSPI` is deployed automatically to the HPOM 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 action-request ticket is created according to the configuration of the SPI for Remedy ARS.

---

**NOTE**

---

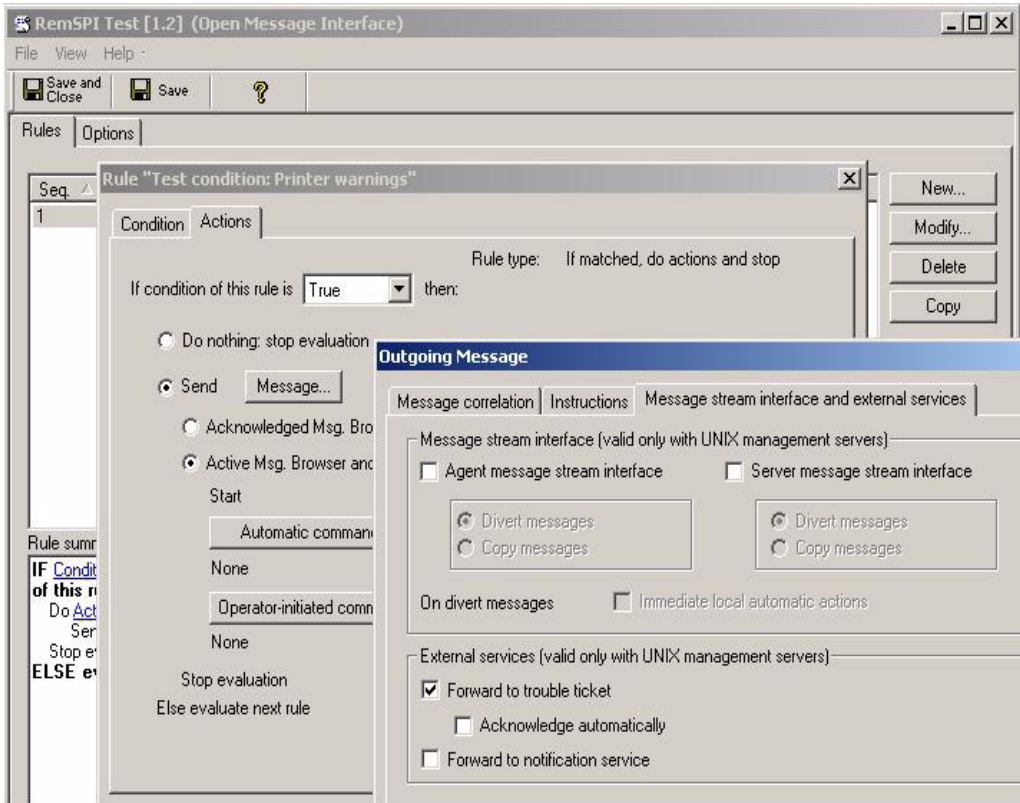
No special user action is required to enable the Trouble Ticket interface on the HPOM management server.

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

1. Enable the Trouble-Ticket interface for the individual c message conditions that will generate messages for ARS:
  - a. In the HPOM 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 HPOM for Unix management servers only. This option will be evaluated by the SPI for Remedy ARS, not the HPOM 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 6-1 Activating automatic forwarding to the Trouble Ticket interface in HPOM



## Customizing the Rules File

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

- “The Structure of the Rules File”
- “Defining Global Options”
- “Defining Targets”
- “Defining Buffer Rules”
- “Defining Update Rules”
- “Defining Submit and Attach Rules”
- “Defining Move-From and Move-To Rules”
- “Defining Conditions”
- “Defining Actions”
- “An Example of a Rules File”

---

### NOTE

After making changes to the `rules` file, you need to apply the changes by saving the file and stopping and restarting the SPI for Remedy ARS server process, which runs on the HPOM management server.

---

## The Structure of the Rules File

The `rules` file is a plain text file, which resides in the directory `%OvShareDir%\conf\remspi`. By default, the `rules.PR` 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 action-request servers and schemas
- update rules that determine how the SPI for Remedy ARS updates an HPOM message if the corresponding action request changes

- submit/attach rules that determine how the SPI for Remedy ARS updates an HPOM message if the message is submitted as/attached to an action-request ticket
- move from/to rules that determine how the SPI for Remedy ARS updates an HPOM message if the action-request ticket is moved away/to a target by means of an action-request transfer (DSO)
- conditions that filter the HPOM messages submitted to the SPI for Remedy ARS
- actions that determine how to create or update action requests when an HPOM message satisfies a defined condition

Figure 6-2

### The Structure of the Rules File

```
File Description
Global Options
Target 1
  Buffer Rules
  Update Rules
  ...
Target n
  Buffer Rules
  Update Rules

Condition 1
  Action
Condition 2
  Action
...
Condition n
  Action
```

The main body of the `rules` file, as illustrated in [Figure 6-2](#), 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"`

**Customizing the Rules File**

- 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
```

- Text strings, which are not keywords, are always enclosed in quotation marks, e.g. "MyTargetName". This rules applies even if the string does not contain embedded white space.
- 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 6-17 on page 129](#).

**Defining Global Options**

A *global option* controls the overall behavior of the SPI for Remedy ARS. [Table 6-6](#) shows the option names that you can define globally.

**Table 6-6**

**Global Options**

Option	Description
NO_AR_UPDATE	The SPI for Remedy ARS does not process changes to HPOM messages. Changing an HPOM message does not result in a change to the associated action-request ticket.

Table 6-6

Global Options (Continued)

Option	Description
ANNO_FAILED_MSG_UPD	<p>If an HPOM message could not be updated as a consequence of a change of the associated action-request ticket, an annotation is added to this HPOM message.</p> <p>Note that this added annotation does not trigger an update of the associated action-request ticket.</p> <p>This option is valid for message updates specified in the ITO_UPDATE and ON_SUBMIT sections of the rules file</p>
BUFFER_TT	<p>This option controls whether the ticket creation and update requests should be buffered if the action-request server is down or not available.</p>
FORGET_ACKED_MSGS	<p>The SPI for Remedy ARS maintains its own database containing the IDs of messages and tickets. If an HPOM message is acknowledged, the entries in the SPI database are deleted if this option is set. This helps to clean up the database and speeds up the processing of the SPI for Remedy ARS. However, if the HPOM message is unacknowledged later, it cannot be updated by the SPI for Remedy ARS.</p>

**Table 6-6 Global Options (Continued)**

Option	Description
FLEXIBLE_TICKET_UPDATES	<p>If this option is <i>not</i> set, an existing AR ticket is updated - if the corresponding HPOM message is modified - via the <i>same</i> SPI rules file condition which was used to <i>create</i> the ticket.</p> <p>If this option <i>is</i> set and an HPOM message is modified, the rules file condition used to update the corresponding AR ticket is determined by checking all rules file conditions from the beginning until a match is found - similar to the process used when creating the ticket. New condition terms which may be used in such update conditions are MESSAGE_STATUS (the status of the modified HPOM message) and CREATE_CONDITION (the name of the condition which was used to create the corresponding ticket). Note that for conditions used solely for updating, only fields marked as UPDATEable make sense. If this option is set and no matching rules file condition is found for a modified HPOM message, the AR ticket is not updated.</p>
ANNO_SEPARATOR "string"	<p>This option defines the separator string in case multiple OM message annotations are concatenated for a ticket field by using the action variable \$MSG_ANNO\$. The string may consist of multiple lines and thus may include line breaks.</p>

The following example defines all global options.

```
OPTION NO_AR_UPDATE
```



```
OPTION ANNO_FAILED_MSG_UPD
OPTION BUFFER_TT
OPTION FORGET_ACKED_MSGS
OPTION FLEXIBLE_TICKET_UPDATES
OPTION ANNO_SEPARATOR "..."
```

If buffering is turned on (OPTION BUFFER\_TT), then the requests for creating and updating action-request tickets are buffered until the action-request server is available again.

If the action-request 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 HPOM messages are not buffered.

If you need more detailed messages regarding the buffering status, use the variable REMSPI\_QUEUE\_CHECK in file `remspi.cfg`. The value of this variable is treated as a time delay after which the number of buffered requests is checked. If the number of buffered requests is greater than zero, an internal warning message is written to the SPI's log file - and forwarded to the HPOM message browser. Note that this check is performed and a message is generated on each check cycle - if a buffered request exists. The number of requests existing in the buffer queues is added to the message. If the variable is not specified or set to 0, no check is performed.

See [“Defining Buffer Rules” on page 85](#) for details regarding the definition of rules for buffering.

## Defining Targets

A *target* defines the name, location of a form (schema) on an action-request 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 action-request server and create or change action requests that belong to a particular schema.

**Table 6-7** Target Attributes

Attribute	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 (primary) action-request server runs. For example, <i>ars1.yourcompany.com</i> . This value is <i>mandatory</i> .
BACKUP_SERVER	The name of the machine where the backup action-request server runs. For example, <i>ars1.yourcompany.com</i> . This value is <i>optional</i> . If the primary server is not reachable, the ticket is created on the backup server - if configured.
SCHEMA	The name of the form (schema). <i>This attribute is mandatory.</i>
USER	The user name with which the SPI for Remedy ARS server logs in to the action-request server. <i>This attribute is mandatory.</i>  Note that you should use different account names for the SPI for Remedy ARS server logging in to the action-request server and for interactive users to be able to distinguish between transactions triggered by the SPI for Remedy ARS and transactions triggered by interactive users.

**Table 6-7 Target Attributes (Continued)**

Attribute	Description
PASSWORD	<p>The encrypted password for USER.</p> <p>Note that this password must be encrypted with the command <code>remspipasswd</code>.</p> <p>If this attribute is not specified, it defaults to the blank (empty) password.</p>
ARTCPPORT	<p>The TCP port on which the action-request server is running. Note that this is only necessary if you configured the action-request server to run on a fixed port and not using the portmapper. If this optional attribute is missing or specified with a value of 0, then the portmapper is used from the SPI for Remedy ARS server. See “Communicating with the Action-Request Server through a Firewall” on page 56 for more details.</p>
ARRPC	<p>The number of the dedicated server queue the SPI for Remedy ARS server should use for communication with the action-request server. If this optional attribute is missing or a value of 0 is specified, then the fast server queue is used. See “Using Private Queues on the Action-Request Server” on page 56 for more details.</p>
AR_SESS_TIMEOUT_NORMAL	<p>An integer value for the timeout in seconds used in fast AR server operations. If this optional attribute is not defined, then the global default value defined in the <code>remspi.cfg</code> is used.</p>
AR_SESS_TIMEOUT_LONG	<p>An integer value for the timeout in seconds used in list AR server operations. If this optional attribute is not defined, then the global default value defined in the <code>remspi.cfg</code> is used.</p>

**Table 6-7 Target Attributes (Continued)**

Attribute	Description
AR_SESS_TIMEOUT _XLONG	An integer value for the timeout in seconds used in definition updating and import/export operations on the AR server. If this optional attribute is not defined, then the global default value defined in the <code>remspi.cfg</code> is used.
ADDITIONAL USER "User-Name" PASSWORD "Encr-Password"	An additional AR user account, which may be used for operations on AR tickets. Multiple additional users may be defined by repeating this specification.

Table 6-7 shows the attributes that you can define for a target. Each attribute must appear once per target definition (ARTCPPORT and ARRPC are optional). 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`. If this primary server is not available, the backup server `ars2.bbn.hp.com` is contacted. If the backup server is not available either, the request is queued (if the OPTION BUFFER\_TT is set). The default communication is explicitly specified.

```
TARGET "ProbSys"
  SERVER "ars1.bbn.hp.com"
  BACKUP_SERVER "ars2.bbn.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "c8D5F221C679B2DA"
  ARTCPPORT 0
  ARRPC 0
  ADDITIONAL USER "ar_user_1" PASSWORD "c83d374274658"
  ADDITIONAL USER "ar_user_2" PASSWORD "33da736492d"
```

---

**NOTE**

To display the encrypted version of a password, use the command:  
`remspipasswd <password>`.

---

## Defining Buffer Rules

Each target has a set of buffer rules. The buffer rules define which priority should be attached to buffered requests for HPOM messages. These rules only apply if the global `OPTION BUFFER_TT` is turned on.

Three priorities are available: `HIGH`, `MEDIUM` and `LOW`. If a request has to be buffered, the SPI for Remedy ARS checks the severity of the relevant HPOM message and stores the request in the appropriate priority queue.

```
BUFFER
  PRIO HIGH      severity ...
  PRIO MEDIUM   severity ...
  PRIO LOW       severity ...
  PRIO DISCARD  severity ...
```

The severity of a message is specified using the keywords:

- Critical
- Major
- Minor
- Warning
- Normal
- Unknown

Multiple severities may be specified for a priority queue. If a severity is not specified, the priority of the queue defaults to `LOW`, that is: the message is stored in the `LOW` priority queue. If the complete `BUFFER` section is omitted, all requests are stored in the `LOW` priority queue. The special priority `DISCARD` may be used to skip buffering requests with specific severities.

The retry intervals for the priority queues may be adjusted in the `remspi.cfg` file using the following syntax:

```
REMSPI_QUEUE_READ_HIGH interval
REMSPI_QUEUE_READ_MEDIUM interval
REMSPI_QUEUE_READ_LOW interval
```

The *interval* is the delay in seconds between two checks of the relevant queue. The default values are as follows:

- `REMSPI_QUEUE_READ_HIGH 300`

## Customizing the Rules File

- REMSPI\_QUEUE\_READ\_MEDIUM 600
- REMSPI\_QUEUE\_READ\_LOW 900

---

**NOTE**

Realtime events arriving during queue checks are processed immediately.

---

### 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 HPOM messages when it receives details of modifications to an action request.

---

**NOTE**

The update rules are also evaluated if the action-request ticket is moved between action-request servers or forms (DSO).

---

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 6-8 on page 87](#) 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”
- “Starting an Operator-Initiated Action”
- “Taking Ownership of a Message” (disowning first on request)
- “Removing Ownership from a Message”
- “Modifying the text of a Message”
- “Modifying the severity of a Message”
- “Modifying or Adding a Custom Message Attribute”

---

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

---

The special manually constructed action-request filter mentioned in [Table 6-8 on page 87](#) is described in “[Configuring the IF\\_MODIFIED ARS Filter manually for itupdate](#)” on [page 145](#).

If the keyword IF\_MODIFIED is used, the action-request filter which notifies the SPI for Remedy ARS server must be constructed in a special way. This must be done manually using the action-request administrator’s GUI, since the program `remspifilter` is not able to generate those filters. To learn about these special action-request filters, see [Configuring the IF\\_MODIFIED ARS Filter manually for itupdate](#) on [page 145](#).

**Table 6-8 Update Keywords**

Keyword	Parameters	Description
ACKNOWLEDGE	[IF_MODIFIED]  <i>field ID</i>  <i>value</i>	Use this keyword to acknowledge the HPOM message if the <i>field ID</i> has the given <i>value</i> . This also removes the association between the action request and the HPOM message from the SPI for Remedy ARS database.  If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.

**Table 6-8 Update Keywords (Continued)**

Keyword	Parameters	Description
UNACKNOWLEDGE	[IF_MODIFIED] <i>field ID</i> <i>value</i>	<p>Use this keyword to unacknowledge the HPOM message if the <i>field ID</i> has the given <i>value</i>. This action is valid only if the message was locally acknowledged using the HPOM GUI.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>
ANNOTATE	[ IF_MODIFIED <i>field ID</i> ] <i>format</i>	<p>Use this keyword to add an annotation to the HPOM message. The <i>format</i> describes the annotation to the HPOM message and can consist of text and variables.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>



**Table 6-8 Update Keywords (Continued)**

Keyword	Parameters	Description
ANNOTATE	[IF_MODIFIED] <i>&lt;field ID value&gt;</i> <i>format</i>	<p>Use this keyword to add an annotation to the HPOM message if the <i>field ID</i> has a special <i>value</i>. The <i>format</i> describes the annotation to the HPOM message and can consist of text and variables.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>
OP_ACTION	<i>command name</i>	<p>Use this keyword to start an operator-initiated action if the SPI for Remedy ARS client passes the given <i>command name</i>.</p>
OWN	[IF_MODIFIED] <i>field ID</i> <i>value</i>	<p>Use this keyword to take ownership of the HPOM message if the <i>field ID</i> has the given <i>value</i>.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>

**Table 6-8** Update Keywords (Continued)

Keyword	Parameters	Description
FORCE OWN	[IF_MODIFIED]  <i>field ID</i>  <i>value</i>	<p>Use this keyword to remove the ownership of the message and then to take ownership of the HPOM message if the <i>field ID</i> has the given <i>value</i> (regardless which user currently owns the message).</p> <p>Note that the REMSPI_ITO_ADMIN_PASSWD keyword is necessary in the <code>remspi.cfg</code> file for this to work. This applies on HPOM for Windows, too, but any password may be used.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>

**Table 6-8 Update Keywords (Continued)**

Keyword	Parameters	Description
DISOWN	[IF_MODIFIED] <i>field ID</i> <i>value</i>	<p>Use this keyword to remove ownership of the HPOM message if the <i>field ID</i> has the given <i>value</i> (regardless which user currently owns the message).</p> <p>Note that to use this keyword, the password for the HPOM administrator has to be configured with the REMSPI_ITO_ADMIN_PASSWD keyword in the <code>remspi.cfg</code> file. This applies on HPOM for Windows, too, but any password may be used.</p> <p>If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an action-request user transaction and the action-request filter was constructed in a special way.</p>
MSGTXT	[IF_MODIFIED] <i>field ID</i> <i>value</i> <i>format</i>	<p>Use this keyword to modify the text of the HPOM message if the <i>field ID</i> has a special <i>value</i>. The format describes the new message text of the HPOM 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.</p>

**Table 6-8 Update Keywords (Continued)**

Keyword	Parameters	Description
SEVERITY	[IF_MODIFIED] <i>field ID</i> <i>value</i> <i>severity</i>	Use this keyword to modify the severity of the HPOM message if the <i>field ID</i> has a special value. The <i>severity</i> describes the new message severity of the HPOM message (Critical, Major, Minor, Warning, Normal). If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> was modified in an AR user transaction and the AR filter was constructed in a special way.
CMA	[IF_MODIFIED] <i>field ID</i> <i>value</i> <i>cma-name</i> <i>format</i>	Use this keyword to add or modify the contents of a custom message attribute with name <i>cma-name</i> of the HPOM message if the <i>field ID</i> has a special <i>value</i> . The <i>format</i> describes the new contents of the CMA and can consist of text and variables.  If the optional keyword IF_MODIFIED is used, this update rule is triggered only if the <i>field ID</i> 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 HPOM 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 HPOM message if the modified action request has the value `Closed` in field 7 and this field was modified in the current action-request transaction.

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

---

**NOTE**

Note that if the `IF_MODIFIED` tag is used in the update rules, the action-request filters which trigger the notification of the SPI for Remedy ARS server must be constructed manually in a special way. The program `remspifilter` is not able to generate these filters on the action-request server. To learn about these special action-request filters, see [Configuring the IF\\_MODIFIED ARS Filter manually for itupdate on page 145](#).

---

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 185](#)

### Unacknowledging Messages

The following example shows an update rule that unacknowledges the HPOM 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 "C338D5F22121E076C200F6E6B4E94B2C679B2DA"
  ITO_UPDATE
    ANNOTATE "$5$ has modified the action request.\\n He or she
made the following work log entry: \\n\\n$1000000001$"
```

Customizing the Rules File

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 HPOM message only if the field 7 was modified in the current action-request transaction by an interactive action-request user. For this example to execute properly, the action-request filters which trigger the notification of the SPI for Remedy ARS server and thus the evaluation of the update rules, must be constructed manually in a special way. The program `remspifilter` is not able to create these filters.

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

You can also use annotation variables in the format that follows an ANNOTATE keyword. [Table 6-9](#) shows the annotation variables that you can use. These variables must be embedded in dollar signs. In addition to the variables listed, you may also refer to the contents of AR ticket fields by embedding the field number in dollar signs. More, all attributes of the current HPOM message may be used ([Table 6-13 on page 113](#)).

Table 6-9

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 action-request 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 $ARS_ID$.
```

### Starting an Operator-Initiated Action

The next example shows an update rule that starts the HPOM 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 for itupdate" on page 147](#).

```
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 HPOM 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 HPOM 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"
```

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 185](#)

### Removing Ownership from a Message

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

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

### Modifying the text of a Message

The following example shows an update rule that use the MSGTXT keyword. The rule instructs the SPI for Remedy ARS to modify the text of the corresponding HPOM message if the modified action request has the value `Rejected` in field 7. The new text of the message contains information about the ticket ID, the rejecting user and the original message text.

You can also use annotation variables in the format that follows a MSGTXT keyword. [Table 6-9 on page 94](#) shows the annotation variables that you can use. These variables must be embedded in dollar signs. In addition to the variables listed, you may also refer to the contents of AT ticket fields by embedding the field number in dollar signs. You can use all the attributes of the current HPOM message as described in [Table 6-13 on page 113](#).

```
ITO_UPDATE
  MSGTXT 7 "Rejected" "Ticket $ARS_ID$ rejected by $5$
($MSG_TEXT_ORIG$)"
```

### Modifying the severity of a Message

The following example shows an update rule that uses the SEVERITY keyword. The rule instructs the SPI for Remedy ARS to set the severity of the corresponding HPOM message to `Minor` (keyword, i.e. no quotes) if the modified action request has the value `Minor` in field `10000002` (string, i.e. with quotes). More than one SEVERITY rule may be specified, with different conditional values.

```
ITO_UPDATE
  SEVERITY 10000002 "Minor" Minor
```

### Modifying or Adding a Custom Message Attribute

The following example shows an update rule that use the CMA keyword. The rule instructs the SPI for Remedy ARS to add or to modify the CMA with name `CMA-1` of the corresponding HPOM message if the modified action request has the value `Rejected` in field 7. The text of the CMA contains information about the ticket ID, the rejecting user and the message text.

You can also use annotation variables in the format that follows a MSGTXT keyword. [Table 6-9 on page 94](#) shows the annotation variables that you can use. These variables must be embedded in dollar signs. In addition to the variables listed, you may also refer to the contents of action-request ticket fields by embedded the field number in dollar signs. More, all attributes of the current HPOM message may be used ([Table 6-13 on page 113](#)).



```
ITO_UPDATE
  CMA 7 "Rejected" "CMA-1" "Ticket $ARS_ID$ rejected by $5$
($MSG_TEXT_ORIG$)"
```

## Defining Submit and Attach Rules

Each target has a set of submit and attach rules. The submit rules define how the SPI for Remedy ARS server updates the HPOM messages when it submits these messages as an action-request ticket. The attach rules define how the SPI for Remedy ARS server updates the HPOM messages when it attaches messages to an existing action-request ticket. You define the submit rules by adding the ON\_SUBMIT attribute to the target definition. The attach rules are defined by adding the ON\_ATTACH 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”
- “Setting or Modifying a Custom Message Attribute (CMA) of a Message”
- “Generating an HPOM message”
- “Adding an annotation”

You then assign keywords with parameters to the ON\_SUBMIT resp. ON\_ATTACH attribute. [Table 6-10 on page 97](#) shows the keywords you use to define for a target.

**Table 6-10**      **Submit and Attach Keywords**

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

**Table 6-10**                      **Submit and Attach Keywords (Continued)**

<b>Keyword</b>	<b>Parameters</b>	<b>Description</b>
OWN	-	Use this keyword to take ownership of the HPOM message right after the message is submitted as an action-request ticket.
FORCE OWN	-	Use this keyword to remove the ownership of the message and then to take ownership of the HPOM message right after the message is submitted as an action-request 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 HPOM message right after the message is submitted as an action-request ticket.  Note that to use this keyword, the password for the HPOM 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 HPOM message text (if specified).

**Table 6-10**                      **Submit and Attach Keywords (Continued)**

Keyword	Parameters	Description
MSGTXT_PREFIX	A string which will be prepended to the original message text. See <a href="#">“Modifying the text of a Message” on page 96</a> for details.	Use this keyword to modify the text of the HPOM message right after the message is submitted as an action-request ticket. Note that after modification, the message is automatically owned by the HPOM core system. You can disown the message immediately after the modification by using the DISOWN submit keyword. An annotation documenting the modification is automatically added by the HPOM core system.
CMA	First string parameter is the name of the CMA to be set, second string parameter the value of the CMA. See <a href="#">“Setting or Modifying a Custom Message Attribute (CMA) of a Message” on page 102</a> for details.	Use this keyword to set or modify a custom message attribute of the HPOM message right after the message is submitted as an AR ticket. Note that after modification, the message is automatically owned by the HPOM core system. You can disown the message immediately after the modification by using the DISOWN submit keyword. An annotation documenting the modification is automatically added by the HPOM core system.

**Table 6-10**                      **Submit and Attach Keywords (Continued)**

Keyword	Parameters	Description
MSG_IF_MANUAL	<p>A string which is used as the message text.</p> <p>See “<a href="#">Generating an HPOM message</a>” on <a href="#">page 103</a> for details.</p>	<p>Use this keyword to specify the text of the HPOM message which is generated right after the current message is successfully and manually submitted as an action-request ticket (\$ITO_SUBMITTER\$ is not "OpC").</p> <p>The new HPOM message has the following attributes:</p> <ul style="list-style-type: none"> <li>• Application "Remedy SPI"</li> <li>• Object "RemSPI"</li> <li>• Message Group "Remedy-SPI"</li> <li>• Severity "Normal"</li> </ul>
ANNO_IF_MANUAL	<p>A string which is used as the annotation text.</p> <p>See “<a href="#">Adding an annotation</a>” on <a href="#">page 103</a> for details.</p>	<p>Use this keyword to specify the text of the annotation which is attached to the HPOM message right after the message is submitted as an action-request ticket.</p> <p>The annotation is added only for successfully and manually submitted messages. If multiple messages are submitted, the annotation is added to all messages.</p>

**Acknowledging Messages**

The following example shows a submit rule that acknowledges the HPOM message after the message is submitted as an action-request 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 185](#).

### 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 HPOM message (regardless which user owns the message currently) if the message is submitted as an action-request 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

If you use the `FORCE` keyword, you have to configure the password of the HPOM administrator in the `remspi.cfg` configuration file using the `REMSPI_I_TO_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 185](#).

### 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 HPOM message if the message is submitted as an action-request ticket.

```
ON_SUBMIT
DISOWN
```

---

#### NOTE

If you use the `FORCE` keyword, you have to configure the password of the HPOM administrator in the `remspi.cfg` configuration file using the `REMSPI_I_TO_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 action-request ticket to the text of the corresponding HPOM message if the message is submitted as an action-request ticket.

The formatted text may contain the special variables \$ARS\_ID\$, \$TICKET\_SERVER\$, \$BACKUP\_USED\$, references to the current message attributes listed in [Table 6-13 on page 113](#), as well as references to fields of the constructed action-request ticket (use \$<fieldno>\$).

```
ON_SUBMIT
  MSGTXT_PREFIX "ID: $ARS_ID$, Ticket-ID: $100002$ - "
```

---

#### 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.

---

### Setting or Modifying a Custom Message Attribute (CMA) of a Message

The following example shows a submit and an attach rule that uses the CMA keyword. The rule instructs the SPI for Remedy ARS to set the CMAs ARServer and ARTicket of the corresponding action-request ticket if the message is submitted as an action-request ticket or attached to an existing action-request ticket.

The formatted text may contain the special variables \$ARS\_ID\$, \$TICKET\_SERVER\$, \$BACKUP\_USED\$ and references to the current message attributes listed in [Table 6-13 on page 113](#), as well as references to fields of the constructed action-request ticket (use \$<fieldno>\$).

```
ON_SUBMIT
  CMA "ARServer" "$TICKET_SERVER$"
  CMA "AR-ID" "$ARS_ID$"
  CMA "Ticket-ID" "$100002$"

ON_ATTACH
  CMA "ARServer" "$TICKET_SERVER$"
  CMA "AR-ID" "$ARS_ID$"
  CMA "Ticket-ID" "$100002$"
```

### Generating an HPOM 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 HPOM message if a message is successfully and manually submitted as an action-request ticket. The submitter must not be "OpC".

The formatted text may contain the special variables `$ARS_ID$`, `$TICKET_SERVER$`, `$BACKUP_USED$` and references to the current message attributes listed in [Table 6-13 on page 113](#), as well as references to fields of the constructed action-request ticket (use `$(fieldno)$`).

```
ON_SUBMIT
  MSG_IF_MANUAL "AR ticket created with ID $100002$"
```

### 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 HPOM message which is submitted as an action-request ticket. The annotation is sent only if the ticket was successfully and manually submitted. .

The formatted text may contain the special variables `$ARS_ID$`, `$TICKET_SERVER$`, `$BACKUP_USED$` and references to the current message attributes listed in [Table 6-13 on page 113](#), as well as references to fields of the constructed action-request ticket (use `$(fieldno)$`).

```
ON_SUBMIT
  ANNO_IF_MANUAL "AR ticket created with ID $100002$"
```

### Defining Move-From and Move-To Rules

In order to support the modification of HPOM messages in case an action-request ticket is moved between action-request servers or between forms, each target may have a set of move-from and move-to rules. The move-from rules define how the SPI for Remedy ARS server updates the HPOM messages when a ticket is moved away from the current target. The move-to rules define how the SPI for Remedy ARS server updates the HPOM messages when an action-request ticket is moved to the current target. You define the move-from rules by adding the `ON_MOVE_FROM_ME` attribute to the target definition. The move-to rules are defined by adding the `ON_MOVE_TO_ME` attribute to the target definition.

You may use the same keywords as in submit and attach rules.

If ticket fields are referenced in move-to rules, they refer to the ticket after the move has been processed by the action-request server.

---

**NOTE**

It's not possible to reference fields of an action-request ticket which is moved away from the current target. This is because at the time the SPI for Remedy ARS server learns about the move, the ticket move has already been processed by the action-request server and is thus not available anymore using the old data.

---

## Defining Conditions

Conditions filter the messages that HPOM 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 an HPOM 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 an HPOM 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.

---

---

**NOTE**

Condition descriptions must be unique throughout file. The description of a condition serves as a key reference.

---

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 HPOM message. The keyword `EXTERNAL` in the test part of a condition refers to an external value file, not to a message attribute.

**Table 6-11 Condition Keywords**

Keyword	Description
AA_STATUS	<p>The status of the automatic action of the HPOM message.</p> <p>Syntax:  <code>AA_STATUS [NOT] <i>status</i></code></p> <p>where <i>status</i> one of UNDEFINED, AVAILABLE, STARTED, FINISHED, FAILED</p>
APPLICATION	<p>The application string attached to the HPOM message.</p> <p>Syntax:  <code>APPLICATION [NOT] [LIKE] <i>string</i></code></p>
APPLICATION IN	<p>A file is specified as the parameter. The current HPOM message application attribute is checked for existence in this file and the condition is satisfied if the value is found. The file holds only strings without blanks, one per line, and is plain text.</p> <p>Syntax:  <code>APPLICATION [NOT] IN <i>file-string</i></code></p>
CMA	<p>This keyword needs two strings as parameters. The first parameter is the name of an HPOM Custom Message Attribute, the second parameter the expected value of this CMA.</p> <p>This condition term is true if the HPOM message has a Custom Message Attribute with that name and value.</p> <p>Syntax:  <code>CMA <i>key-string</i> [NOT] [LIKE] <i>val-string</i></code></p>

**Table 6-11 Condition Keywords (Continued)**

Keyword	Description
CMA IN	<p>A CMA name and a file are specified as the parameters. The named custom message attribute of the current HPOM message is checked for existence in this file and the condition is satisfied if the value is found. The file holds only strings without blanks, one per line, and is plain text.</p> <p>Syntax:  <i>CMA key-string</i> [NOT] IN <i>file-string</i></p>
CREATE_CONDITION	<p>The name of the rules file condition which was used to create the AR ticket.</p> <p>This keyword is used in update conditions which should be triggered if an already submitted HPOM message is modified, and the AR ticket should be updated using a different condition than was used to create it.</p> <p>Valid only if the global option FLEXIBLE_TICKET_UPDATES is set in the rules file.</p> <p>Syntax:  CREATE_CONDITION [NOT] <i>name-string</i></p>
DUPLICATES	<p>The number of duplicates of the HPOM message.</p> <p>Syntax:  DUPLICATES <i>op number</i></p> <p>where op is one of: ==, !=, &gt;=, &lt;=, &gt;, &lt;</p>
EXTERNAL	<p>A value in an external value file. For more details, see <a href="#">“Referencing External Values” on page 169</a>.</p> <p>Note that this is keyword behaves different than others in this list. This keyword does not refer to an attribute of the HPOM message, but to a value in an external file.</p> <p>Syntax:  EXTERNAL <i>key-str</i> [NOT] [LIKE] <i>val-str</i></p>

**Table 6-11 Condition Keywords (Continued)**

Keyword	Description
MESSAGE_STATUS	<p>The type of the modification (<i>Message Change Event</i>) of an already submitted HPOM message. Valid only if the global option FLEXIBLE_TICKET_UPDATES is set in the rules file.</p> <p>Syntax:  MESSAGE_STATUS [NOT] <i>status</i>...</p> <p>where <i>status</i> is one of the following:</p> <ul style="list-style-type: none"> <li>• MSG_EVT_CREATE new message arrived</li> <li>• MSG_EVT_ANNO annotations were added</li> <li>• MSG_EVT_ACK acknowledged</li> <li>• MSG_EVT_UNACK unacknowledged</li> <li>• MSG_EVT_OWN owned</li> <li>• MSG_EVT_DISOWN disowned</li> <li>• MSG_EVT_AA_START auto action started</li> <li>• MSG_EVT_AA_END auto action finished</li> <li>• MSG_EVT_OA_START oper action started</li> <li>• MSG_EVT_OA_END oper action finished</li> <li>• MSG_EVT_MODIFY modified</li> <li>• MSG_EVT_CHGSEV severity modified</li> <li>• MSG_EVT_MSGCHG message text modified</li> <li>• MSG_EVT_CMA_UPDATE CMA modified</li> <li>• MSG_EVT_DUPL_RECEIVED duplicate received</li> <li>• MSG_EVT_ANNO_ADD annotation added</li> <li>• MSG_EVT_ANNO_SET annotation modified</li> <li>• MSG_EVT_ANNO_DEL annotation deleted</li> </ul> <p>Multiple <i>status</i> names may be used, separated by blanks.</p>

**Table 6-11 Condition Keywords (Continued)**

Keyword	Description
MSGGRP	<p>The message's message group.</p> <p>Syntax:  MSGGRP [NOT] [LIKE] <i>string</i></p>
NODE	<p>The name of the machine that sent the message. (or for which the message was sent).</p>
NODE_IP	<p>The IP address of the machine that sent the message (or for which the message was sent).</p> <p>Syntax:  NODE_IP [NOT] [LIKE] <i>string</i></p>
NODE IN	<p>A file is specified as the parameter. The current HPOM 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.</p> <p>Syntax:  NODE [NOT] IN <i>string</i></p>
OA_STATUS	<p>The status of the operator initiated action linked to the HPOM message.</p> <p>Syntax:  OA_STATUS [NOT] <i>status</i></p> <p>where <i>status</i> one of UNDEFINED, AVAILABLE, STARTED, FINISHED, FAILED</p>
OBJECT	<p>The name of the object that generated the message.</p> <p>Syntax:  OBJECT [NOT] [LIKE] <i>string</i></p>

**Table 6-11 Condition Keywords (Continued)**

Keyword	Description
OBJECT IN	<p>A file is specified as the parameter. The current HPOM message object attribute is checked for existence in this file and the condition is satisfied if the value is found. The file holds only strings without blanks, one per line, and is plain text.</p> <p>Syntax:            OBJECT [NOT] IN <i>file-string</i></p>
OWN_USER	<p>The name of the HPOM user who owns the HPOM message.</p> <p>Syntax:            OWN_USER [NOT] [LIKE] <i>string</i>            OWN_USER [NOT] NOBODY</p>
SERVICE_NAME	<p>The name of the service attached to the HPOM message.</p> <p>Syntax:            SERVICE_NAME [NOT] [LIKE] <i>string</i></p>
SEVERITY	<p>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.</p> <p>Syntax:            SEVERITY [NOT] <i>severity</i></p> <p>where severity is a blank separated list of Unknown, Normal, Warning, Minor, Major, Critical</p>
TEXT	<p>The text that explains the message's cause, i.e. the message text.</p> <p>Syntax:            TEXT [NOT] [LIKE] <i>string</i></p>

Table 6-11 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 HPOM message match the values you give for the keywords, the HPOM 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 HPOM message.

---

The next example shows a suppress condition that filters out all HPOM messages that have the severity `Normal`. From the HPOM 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 to create an action-request ticket, the SPI for Remedy ARS cannot update the tickets which were created with this 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 *Regular Expressions* via the .NET Framework.

---

**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 6-12 on page 111 lists some examples for using regular expressions in conditions.

**Table 6-12 Regular Expressions in Message Conditions**

Syntax	Description
OBJECT LIKE "^[oracle- [0-9] +\$"	Matches if the HPOM message object attribute starts with the string oracle- and ends with a non empty sequence of digits.
TEXT LIKE "ERROR WARNING"	Matches if the HPOM 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. Note too that, for the SPI for Remedy ARS, it is necessary to quote the quote character.

### 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”

### Customizing the Rules File

- [“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 an HPOM 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 targets, repeat `TARGET` further.

The operations on the ARS server are executed on behalf of the AR user account that was specified with the keyword `USER` for the target. If you want the operations to be executed on behalf of a different user account, you have to specify the additional account with the keyword `ADDITIONAL USER` for this target and reference it here with `SET TARGET target name AS USER user-name`.

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. The operations on `ProbSys1` are executed on behalf of the standard `USER` that was defined for `ProbSys1`, the actions on `ProbSys2` are executed on behalf of the `ADDITIONAL USER ars_user_1`.

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

SET

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

TARGET "ProbSys2" AS USER "ars_user_1"
  FIELD 1000000005 "Copy of AR on ProbSys1. Created by SPI."
```

The following data *types* for fields are supported by the SPI for Remedy ARS:

- KEYWORD
- INTEGER



- REAL
- CHAR
- DIARY
- ENUM
- TIME
- BITMASK

Using a field with any other data type will result in a validation error for the according TARGET.

---

**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 HPOM message.

[Table 6-13](#) shows the action variables that you can use. Each time you use an action variable, place \$ before and after the variable name.

**Table 6-13**

**Action Variables**

Variable	Description	Can Be Empty?
<i>AA_ACK_FLAG</i>	Whether this message's automatic action acknowledges the message.	No
<i>AA_ANNO_FLAG</i>	Whether the output of this message's automatic action becomes a new annotation.	No
<i>AA_CMD</i>	The command for this message's automatic action.	Yes

**Table 6-13**                      **Action Variables (Continued)**

<b>Variable</b>	<b>Description</b>	<b>Can Be Empty?</b>
<i>AA_NODE</i>	The node which this message's automatic action runs on.	Yes
<i>AA_STATUS</i>	The status of this message's automatic action.	No
<i>ACK_TIME</i>	The time that the message became acknowledged.	Yes
<i>ACK_USER</i>	The name of the HPOM user who acknowledged the message.	Yes
<i>ANNO_CNT</i>	The number of annotations to this message.	No
<i>ANNO_USER</i>	The name of the HPOM user who added an annotation last.	Yes
<i>ARS_SERVER</i>	The name of the primary ARS server configured for the target used to submit the ticket.	No
<i>BACKUP_SERVER</i>	The name of the backup ARS server configured for the target used to submit the ticket.	Yes
<i>BACKUP_USED</i>	The numeric index of the server on which the ticket is created, with "0" meaning the primary server. The index follows the order of specification in the <code>rules</code> file.	No

**Table 6-13**      **Action Variables (Continued)**

Variable	Description	Can Be Empty?
<i>CMA</i>	<p>This action variable needs the name of a custom message attribute as a parameter. This pair is then replaced by the value of the customer message attribute in the current HPOM message.</p> <p>The following example will insert into action-request field 9 the value of the CMA Orga of the current HPOM message:</p> <p>Field 9 "Org = \$CMA Orga\$"</p>	Yes
<i>DUPLICATES</i>	<p>The number of duplicates of this HPOM message.</p>	No
<i>EXTERNAL</i>	<p>This action variable needs the name of a key in the external file as a parameter. This pair is then replaced by the value assigned to the key in the external value file.</p> <p>The following example will insert into action-request field 8 the value of the key Phone Number in the external value file:</p> <p>FIELD 8 "Phone = \$EXTERNAL PhoneNumber\$"</p>	Yes
<i>HLP_TEXT</i>	<p>The instructions associated with the message.</p>	Yes
<i>HOST_NAME</i>	<p>The host name of the machine where the HPOM management server and SPI for Remedy ARS server run.</p>	No
<i>ITO_NOTIFY</i>	<p>Whether HPOM passed the message to a notification interface.</p>	No

**Table 6-13**                      **Action Variables (Continued)**

<b>Variable</b>	<b>Description</b>	<b>Can Be Empty?</b>
<i>ITO_TT</i>	Whether HPOM passed the message to the trouble ticket-interface.	No
<i>ITO_TT_ACK</i>	Whether HPOM acknowledges this message automatically after it passes it to the trouble-ticket interface.	No
<i>ITO_UNMATCHED</i>	Whether the message did not match a condition on the HPOM agent.	No
<i>ITO_USER</i>	The HPOM user name which the SPI for Remedy ARS used to log in to HPOM.	No
<i>LOG_ONLY</i>	Whether HPOM immediately moves the message to the history database.	No
<i>MSG_ANNO</i>	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
<i>MSG_ANNO_ALL</i>	All annotations which are attached to the message.	Yes
<i>MSG_ANNO_FIRST</i>	The first annotation which is attached to the message.	Yes
<i>MSG_ANNO_LAST</i>	The last annotation which is attached to the message.	Yes
<i>MSG_GRP</i>	The message's message group.	No
<i>MSG_ID</i>	The messages ID.	No
<i>MSG_OBJ</i>	The name of the object that generated the message.	No
<i>MSG_SEVERITY</i>	The message's severity.	No

**Table 6-13 Action Variables (Continued)**

<b>Variable</b>	<b>Description</b>	<b>Can Be Empty?</b>
<i>MSG_SRC</i>	The message's source.	No
<i>MSG_SRC_TYPE</i>	The message's source type.	No
<i>MSG_STATUS</i>	Whether the message is active or in the history database.	No
<i>MSG_TEXT</i>	The text that explains the message's cause.	No
<i>MSG_TEXT_ORIG</i>	The message's original text before any changes.	No
<i>MSG_TYPE</i>	The message's type.	Yes
<i>NODE_APPL</i>	The application that generated the message.	No
<i>NODE_DATE</i>	The date on which the HPOM managed node generated the message.	No
<i>NODE_NAME</i>	The name of the machine that sent the message.	No
<i>NODE_IP</i>	The IP address of the machine that sent the message (in decimal string format, e.g. 192.168.0.12)	No
<i>NODE_TIME</i>	The time at which the HPOM managed node generated the message.	No
<i>OA_ACK_FLAG</i>	Whether this message's operator- initiated action acknowledges the message.	No
<i>OA_ANNO_FLAG</i>	Whether the output of this message's operator-initiated action becomes a new annotation.	No
<i>OA_CMD</i>	The command for this message's operator-initiated action.	Yes

**Table 6-13**      **Action Variables (Continued)**

<b>Variable</b>	<b>Description</b>	<b>Can Be Empty?</b>
<i>OA_NODE</i>	The node which this message's operator-initiated action runs on.	Yes
<i>OA_STATUS</i>	The status of this message's operator-initiated action.	No
<i>OWN_TIME</i>	The time at which a user took ownership of this message.	Yes
<i>OWN_USER</i>	The name of the HPOM user who owns the message.	Yes
<i>SERVICE_NAME</i>	The name of the service attached to the HPOM message.	Yes
<i>SRV_DATE</i>	The date on which the HPOM management server received the message.	No
<i>SRV_TIME</i>	The time at which the HPOM management server received the message.	No
<i>TICKET_SERVER</i>	The name of the action-request server on which the ticket has been created.	No
<i>MANAGEMENT_SERVER</i>	The name of the current HPOM management server system or HPOM package (in a cluster environment).	No

Table 6-13 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 HPOM message into field 1000000003
- the name of the application that generated the HPOM 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 121](#).

---

**Defining Values using Composite Formats**

You can use composite formats to define values in the action request using pre-formatted information from the HPOM message. [Table 6-14](#) shows the composite formats available. The composite format MSG\_EVENT varies automatically, depending on how the HPOM message changed. [Table 6-15](#) shows the formats that can apply for MSG\_EVENT.

**Table 6-14 Composite Formats**

Format Name	Description
MSG_EVENT	A format that varies automatically, depending on the type of change made to the HPO message. <a href="#">Table 6-15</a> 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 6-15 MSG\_EVENT Formats

Event	Format
MSG_EVT_AA_END	“Automatic action of HPOM message \$MSG_ID\$ finished.”
MSG_EVT_AA_START	“Automatic action of HPOM message \$MSG_ID\$ has been started.”
MSG_EVT_ACK	“HPOM message \$MSG_ID\$ has been acknowledged by \$ACK_USER\$.”
MSG_EVT_ANNNO	“HPOM message \$MSG_ID\$ has been annotated by \$ANNO_USER\$.”
MSG_EVT_CREATE	“HPOM message \$MSG_ID\$ has been submitted to the Trouble Ticket IF.”
MSG_EVT_DISOWN	“HPOM message \$MSG_ID\$ has been disowned.”
MSG_EVT_OA_END	“Operator-initiated action of HPOM message \$MSG_ID\$ finished.”
MSG_EVT_OA_START	“Operator-initiated action of HPOM message \$MSG_ID\$ has been started.”
MSG_EVT_OWN	“HPOM message \$MSG_ID\$ has been owned by \$OWN_USER\$.”
MSG_EVT_UNACK	“HPOM message \$MSG_ID\$ has been unacknowledged.”
MSG_EVT_MODIFY	“HPOM message \$MSG_ID\$ has been modified.”
MSG_EVT_CHGSEV	“HPOM message \$MSG_ID\$: Severity changed.”
MSG_EVT_MSGCHG	“HPOM message \$MSG_ID\$: Message text changed.”
MSG_EVT_CMA_UPDATE	“HPOM message \$MSG_ID\$: CMAs updated.”
MSG_EVT_DUPL_RECEIVED	“HPOM message \$MSG_ID\$: Duplicate received.”



**Table 6-15 MSG\_EVENT Formats (Continued)**

Event	Format
MSG_EVT_ANNO_ADD	“HPOM message \$MSG_IDS: Annotation added.”
MSG_EVT_ANNO_SET	“HPOM message \$MSG_IDS: Annotation modified.”
MSG_EVT_ANNO_DEL	“HPOM message \$MSG_IDS: Annotation deleted.”

If an annotation has been added or updated, HPOM will send the events MSG\_EVT\_ANNO and MSG\_EVT\_ANNO\_SET. If an annotation has been deleted, HPOM will send MSG\_EVT\_ANNO and MSG\_EVT\_ANNO\_DEL. If the last annotation of a message has been deleted, HPOM will send MSG\_EVT\_NO\_ANNO and MSG\_EVT\_ANNO\_DEL.

**Setting Default Values**

Some of the action variables in [Table 6-13](#) may contain no value, depending on the HPOM 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 122](#).

---

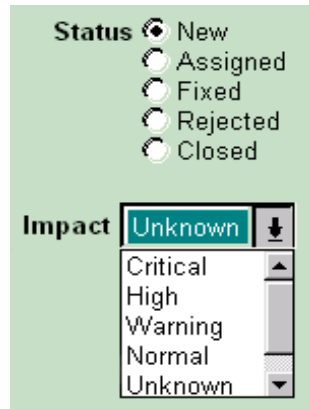
### Setting Enumerated Field Values

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 6-3 shows example enumerated fields in a schema.

Figure 6-3

#### 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 HPOM message do not match the available values in the schema. To do this, you map the index numbers of the HPOM 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 6-3 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 HPOM (0) to unknown (4)
- normal in HPOM (1) to normal (3)
- warning in HPOM (2) to warning (2)
- minor in HPOM (3) to high (1)

- major in HPOM (4) to high (1)
- critical in HPOM (5) to critical (0)

**Setting Update Policies**

By default, the SPI for Remedy ARS does not update any fields in the action request if the HPOM 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 6-16** Update Keywords

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

[Table 6-16](#) 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 HPOM messages are attached to an existing ticket, the annotation of the original HPOM 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:

- HPOM 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 "C338D5F22121E076C200F6E6B4E94B2C679B2DA"
  ITO_UPDATE
  ANNOTATE "Work log entry:\n$1000000001$"

  FIELD 2 `HPOM 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

With the `ATTACH_TO` and `ATTACH_TO_EXISTING` keywords, it is possible to define a list of *search fields* which are then used to locate an action-request ticket, to which the *submitted* HPOM messages should be attached.

If the condition matches, all action-request 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 HPOM messages are attached to the oldest ticket.
- If no tickets are found on a target and the `ATTACH_TO` keyword is used, a new ticket is created as if the HPOM messages were submitted without the search fields.
- If no tickets are found on a target and the `ATTACH_TO_EXISTING` keyword is used, no new ticket is created. The condition is treated as *not matching* and the next condition is checked.

If you want to modify this behavior in such a way that the next condition is *not* checked, i.e. the search for matching conditions would terminate and the HPOM message would effectively be dropped, then you may add a *suppress condition* with the same conditional terms as the next condition. See [Example 6-2 on page 126](#) for details.

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

[Example 6-1 on page 125](#) shows a condition definition with search fields. In this example, every time an HPOM 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 HPOM message.

If such tickets are found, the currently submitted HPOM 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.

#### Example 6-1

#### Automatic attachments with `ATTACH_TO`

```
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 "$MSG_TEXT$"
```

**Example 6-2 Modified behavior for ATTACH\_TO\_EXISTING**

If the ATTACH\_TO\_EXISTING keyword does not find an existing ticket using the list of *search fields*, the condition is treated as *not matching* and the next condition is checked. If you don't want the current HPOM message to be further processed but to be dropped instead, insert a *suppress condition* right after the ATTACH\_TO\_EXISTING condition and use the same *conditional terms* (without the search fields).

```
MSGCONDITIONS  
DESCRIPTION "Attach-To-Existing-Or-Drop-1"  
CONDITION  
SEVERITY Critical  
OBJECT LIKE "ora.*"  
ATTACH_TO_EXISTING  
FIELD 9 "$MSG_OBJ$"  
SET TARGET "ProbSys"  
FIELD 9 "$MSG_OBJ$"  
UPDATE APPEND FIELD 10 "$MSG_TEXT$"
```

```
SUPPRESSCONDITIONS  
DESCRIPTION "Attach-To-Existing-Or-Drop-2"  
CONDITION  
SEVERITY Critical  
OBJECT LIKE "ora.*"
```

```
MSGCONDITIONS  
... More message conditions may follow
```

---

**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 "$MSG_TEXT$"

  TARGET "OtherTarget"
  FIELD 11 "Something"
  FIELD 9 "$MSG_OBJ"
```

## **An Example of a Rules File**

This section uses the rules file supplied with the SPI for Remedy ARS and displayed in [Example 6-3](#) to demonstrate how the field names of an ARS schema shown in [Figure 6-4 on page 128](#), are associated with unique identifiers, shown in [Table 6-17 on page 129](#). This section covers the following topics:

- “Submitting HPOM Messages”
- “Updating an Action Request”
- “Updating an HPOM Message”

### Customizing the Rules File

Example 6-3 illustrates how any changes to the fields you identify may then be linked to actions that need to be carried out in HPOM. The action-request schema shown in Figure 6-4 enables help desk users to create and update details of reported problems.

Figure 6-4

Example Schema

**Submit Window -- Problem Report (sdwin95)**

### Problem Report System

*Case ID*  **Status**  New  
 Assigned  
 Fixed  
 Rejected  
 Closed

**Created By**

*Created On*

**Impact**

*Assigned To*

*Last Modified By*

*Modified On*

**Machine Name**

*Application/Device*

**Short Description**

**Description**

*Work Log*



Table 6-17 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 6-17** 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 6-3 shows a simple rules file that defines how the SPI for Remedy ARS handles HPOM messages and action requests in the Problem Report System.

**Example 6-3** Example Rules File

```
*** File Definition ***

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

*** Target Definition ***
```

Customizing the Rules File

```
TARGET "ProbSys"
  SERVER "ars1.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 $$ on $$"

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

  *** 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

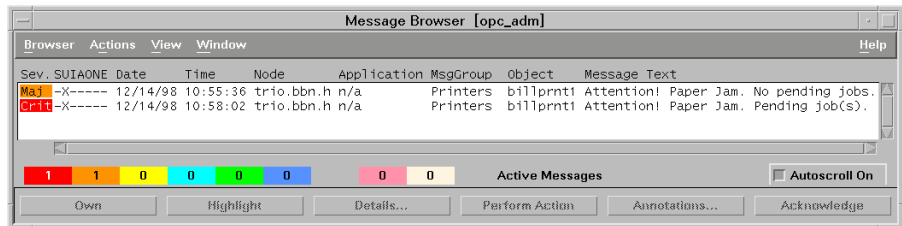
SET TARGET "ProbSys"
  UPDATE ALL FIELD 1000000001 "$MSG_EVENT$"
  FIELD 1000000004 "$NODE_APPL$"
  ALL FIELD 1000000005
"Critical Error. Submitted from ITO.
$MSG_PARMS$
Annotation: $MSG_ANNO$"
    
```

### Submitting HPOM Messages

Figure 6-5 shows two HPOM messages that appear when a large printer fails.

Figure 6-5

### HPOM Messages for Submission



When the HPOM user decides to submit these messages to the action-request system, the following process 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\_SUBMITsection.

Figure 6-6 Created Action Request

The screenshot displays the 'Problem Report' window in Remedy ARS. The window title is 'Problem Report'. It shows a 'Matching Problem Report' section with a 'Short Description' of 'Attention: Paper Jam. Pending Job(s)'. Below this is a 'Modify Problem Report 0000000000000001' section. The 'Case ID' is '0000000000000001', 'Created By' is 'ppc\_admin', 'Create On' is '06.03.00 14:29:52', 'Assigned To' is empty, 'Last Modified By' is 'Demo', 'Modified On' is '06.03.00 14:32:57', 'Machine Name' is 'cholera.nice.de', and 'Application/Device' is 'billprt1'. The 'Status' is 'New', and 'Impact' is 'Unknown'. The 'Short Description' is 'Attention: Paper Jam. Pending Job(s)', and the 'Description' is 'Printing Error. \* This action request is linked to ITO message:'. There is a 'Work Log' field at the bottom. Buttons for 'Start Action' and 'Escalate' are at the bottom of the form.

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

## Customizing the Rules File

- 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 HPOM message.

---

### NOTE

The first condition in [Example 6-3](#) creates a different action request if the HPOM 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 6-3](#) creates an action request for all critical HPOM messages that do not meet the first or second conditions.

---

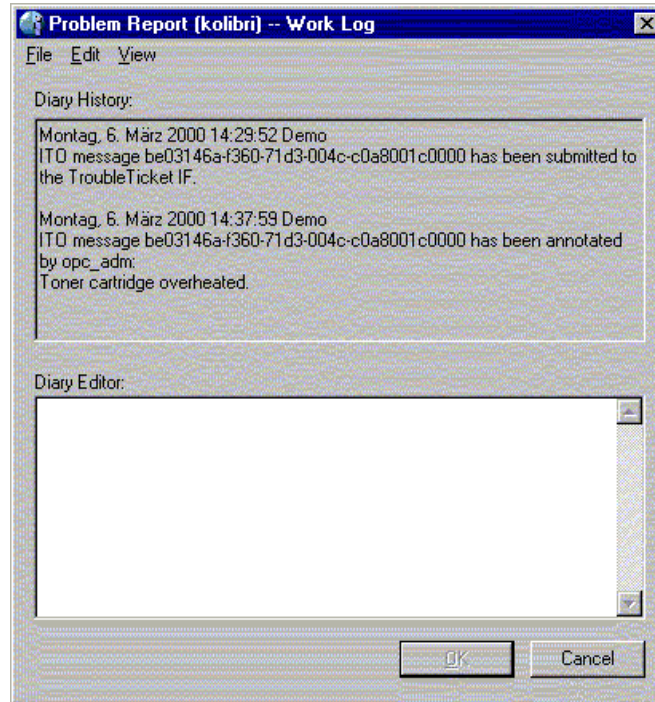
### Updating an Action Request

A HPOM user has experience with this type of large printer, and notices the HPOM message in the message browser. She adds an annotation to the HPOM 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 6-3](#) when creating the action request, it also applies the second condition when updating it.

Figure 6-7

### 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 HPOM user adds the annotation, the SPI for Remedy ARS appends details of this event to the action request's work log. [Figure 6-7](#) shows how the annotation in HPOM has updated the work log entry in the action request.

---

#### NOTE

When the SPI for Remedy ARS applies the first condition in [Example 6-3](#), it cannot update any of the action request's fields.

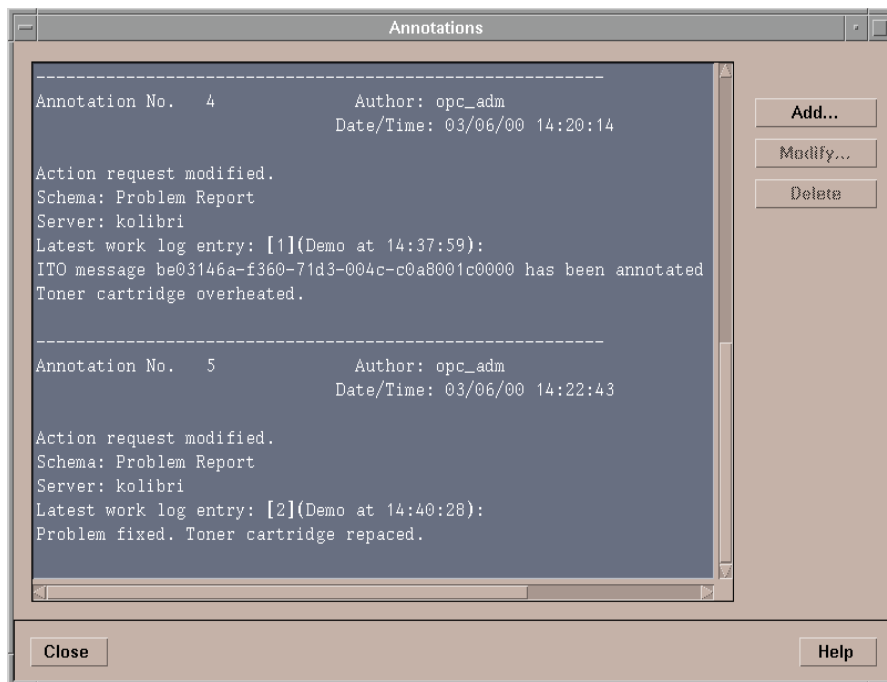
When the SPI for Remedy ARS applies the third condition in [Example 6-3](#), it can only update the `Work Log` field.

---

### Updating an HPOM Message

Figure 6-8 shows how amendments to the Status and Work Log of the action request cause an annotation to be added automatically to the corresponding HPOM message—the same HPOM message, which created the original 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 6-8 Changed Action Request



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

- add an annotation to the HPOM message that includes the Work Log entry
- take ownership of the HPOM 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 HPOM 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.

There are 2 methods to notify the SPI for Remedy ARS server about the change, one starting a new process for each modification, the other using an AR Plugin.

2. Configure active links in ARS which enable users to start actions in HPOM directly from the action-request schema

This section describes the following topics:

- [“Method 1: The itoupdate command”](#)
- [“Method 2: The ItoUpdatePlugin plugin”](#)
- [“The remspifilter command”](#)
- [“Configuring the ARS Filter automatically”](#)
- [“Configuring the ARS Filter manually for itoupdate”](#)
- [“Configuring an Active Link for itoupdate”](#)

---

### NOTE

You should consider creating the action-request filters manually if you want to notify the SPI for Remedy ARS server only if an interactive action request user has changed the ticket. The filters constructed automatically by `remspifilter` execute on any modification of the ticket, regardless whether it was an interactive action-request user or the SPI for Remedy ARS server itself. This can lead to multiple annotations to the HPOM message.

If you are going to use the SPI for Remedy ARS Plugin instead of the traditional SPI for Remedy ARS client `itoupdate`, then you have to configure the according action-request filter manually. The tool `remspifilter` is not able to create the necessary AR filter.

---

## Method 1: The itoupdate command

The action-request 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 action-request 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 4.06” on page 38.

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

- UNIX operating systems:
  - `/var/opt/OV/bin/instrumentation/itoupdate`
- MS Windows operating systems:
  - `%OvAgentDir%\bin\instrumentation\itoupdate.exe`

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

The options `-retries` and `-timeout` may be used to implement a kind of buffering for AR ticket changes. If the `itoupdate` command is not able to connect to the SPI server and to transmit the request, it may retry the attempt after a certain delay.

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

```
itoupdate [opts] <Remedy SPI server hostname> <params> ...
```

The allowed options (*opts*) are as follows, note that the order must not be changed, but all options are *optional*:

<code>-info</code>	log the connection information; automatically enabled in debug mode
<code>-debug</code>	enable the debug mode
<code>-retries <i>n</i></code>	set number of connection retries (Default: 10)
<code>-timeout <i>n</i></code>	set connection retry delay in seconds (Default: 3)
<code>-maxlogfiles <i>n</i></code>	max. number of log files, if the log file size is limited (Default: 1)

-maxlogsize *n*      max. size of log files in bytes (Default: 0 = unlimited)

The allowed parameters (*params*) 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 <code>remspi.cfg</code> 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 `itupdate` command to trigger an action on a HPOM message:

```
itupdate <Remedy SPI server> \  
    ENTRY_ID=<AR ticket id> \  
    SERVER=<AR server name> \  
    SCHEMA=<AR schema name>\  
    CMD=<command name as defined in rules>
```

2. To use the `itupdate` command to inform HPOM about changes to an action-request ticket:

```
itupdate <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 the action-request system by using the action-request variable `$OPERATION$`.

3. To use the `itupdate` command to inform HPOM about a changed field in an action-request ticket:

```
itupdate <Remedy SPI server> \  
    ENTRY_ID=<AR ticket id> \  
    SERVER=<AR server name> \  
    MODIFIED=<field id>
```

```
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 the action-request system by using the action-request variable `$_OPERATION$`. The value of the MODIFIED= parameter is the numeric ID of the field.

4. To use the `itoupdate` command move an action-request ticket from one (source) action-request server to another (destination) action-request 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 a Ticket Between Action-Request Servers](#)” on page 155 for details.

## Method 2: The ItoUpdatePlugin plugin

The action-request filters and active links may notify the SPI for Remedy ARS server by using a plugin, which is loaded into the AR Plugin Server. This way, no external process is started each time the filter or link triggers. Instead, the pre-loaded plugin is used, which is more efficient.

The plugin needs to send exactly the same information to the SPI for Remedy ARS server as the `itoupdate` client in the former method.

The `ItoUpdatePlugin.jar` utility has to be manually installed on the action-request server system in the following location:

- `<AR_SYSTEM_HOME>/Arserver/pluginsvr`

The `ItoUpdatePlugin.jar` utility may be found in the installation directory of the SPI for Remedy ARS binaries on the HPOM management server:

- `<SPI_BIN_DIR>/jplugin`

In addition, the new plugin has to be manually registered with the AR System server.

1. Register the new plugin in the AR configuration file `ar.cfg` by appending the following line (no line breaks). Replace `<AR-Server-Name>` with the name of the AR System server.

```
Server-Plugin-Alias: HP.ARF.ITOUPDATEPLUGIN  
HP.ARF.ITOUPDATEPLUGIN <AR-Server-Name>:9999
```

2. Register the new plugin in the AR configuration file `pluginsvr_config.xml` by inserting the following XML entity into the `<pluginsvr_config>` entity. Replace `<AR_SYSTEM_HOME>` with the complete path of the AR System Server home directory, the filename listed here needs to be the absolute pathname of the `ItoUpdatePlugin.jar` file. The port 13603 is the default port on which the SPI for Remedy ARS server listens for update notifications. If you changed this listening port on the HPOM management server, then you'll have to change the port number here, too.

```
<plugin>  
  <name>HP.ARF.ITOUPDATEPLUGIN</name>  
  <type>FilterAPI</type>  
  <code>JAVA</code>  
  
  <filename><AR_SYSTEM_HOME>/pluginsvr/ItoUpdatePlugin.jar</  
filename>  
  <classname>ItoUpdatePlugin</classname>  
  <userDefined>  
    <remspi_update_port>13603</remspi_update_port>  
  </userDefined>  
</plugin>
```

## The remspifilter command

The `remspifilter` command registers the filters you create in ARS and want to use to perform actions in HPOM. 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 | -ar_aix ] [ -https ]  
             [ -cmd <filter command> ]  
             [ <filter name> ]
```

Where:

## Customizing the SPI for Remedy ARS

### ARS Filters and Active Links

-create	creates a new filter. To retrieve an existing filter, neither specify -create nor -delete.
-delete	deletes the filter you specify in <i>filter</i> name
-target	creates a filter for a <i>target</i> defined in the rules file. Note that differences between target definitions and filter declarations are resolved on the SPI for Remedy ARS server.
-server	defines the name of the action-request 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
-user	defines the user name with which to log in to the action-request server
-passwd	defines a password to be associated with the name defined in -user
-ar_nt	indicates an action-request server running on MS Windows operating systems
-ar_aix	indicates an action-request server running on AIX operating systems
-ar_ux	indicates an action-request server running on UNIX (not AIX) operating systems
-ar_nt	indicates an action-request server running on MS Windows operating systems
-https	indicates an HTTPS agent
-cmd	<i>path of itupdate</i> 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 11.x path; <i>/var/opt/OV/bin/OpC/cmds/</i>
<i>filtername</i>	defines the name of the filter to create, delete, or retrieve. The default <i>filter name</i> is: <i>&lt;Schema Name&gt;:ITO-Update</i>

---

#### 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 the IF\\_MODIFIED ARS Filter manually for itupdate on page 145](#).

In addition, this command is not able to create the filter needed if the `ItoUpdatePlugin` is used.

---

### Configuring the ARS Filter automatically

To automatically configure a filter for the action-request system, use the command `remspifilter`. For example, you could enter the following command:

```
remspifilter -create -cmd \  
<InstallationDir>/itoupdate -server arsl \  
-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:

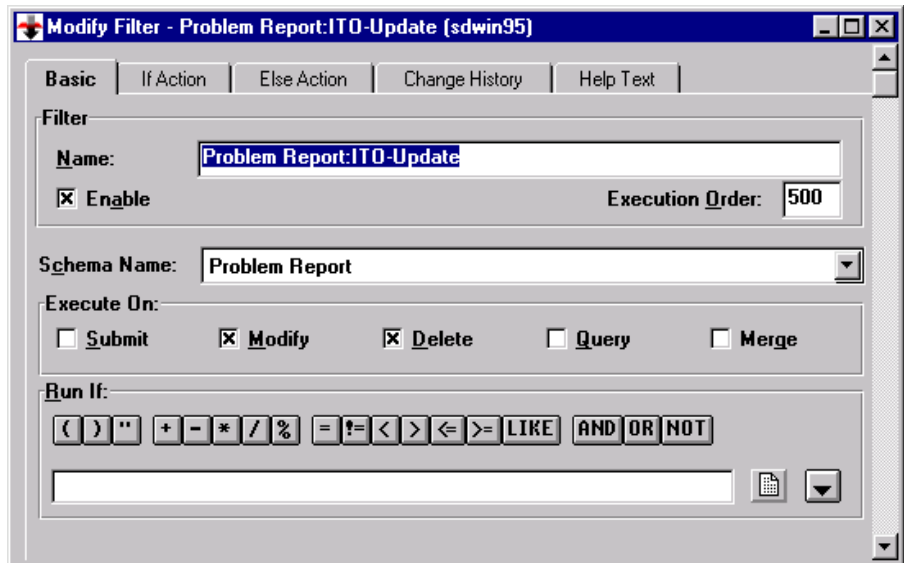
```
remspifilter -create -cmd <InstallationDir>/itoupdate
```

### Configuring the ARS Filter manually for `itoupdate`

You can configure the filter manually in the same way that you create other filters in an action-request system. To do this:

1. Create a new filter that executes when someone modifies or deletes an action request as shown in [Figure 6-9](#).

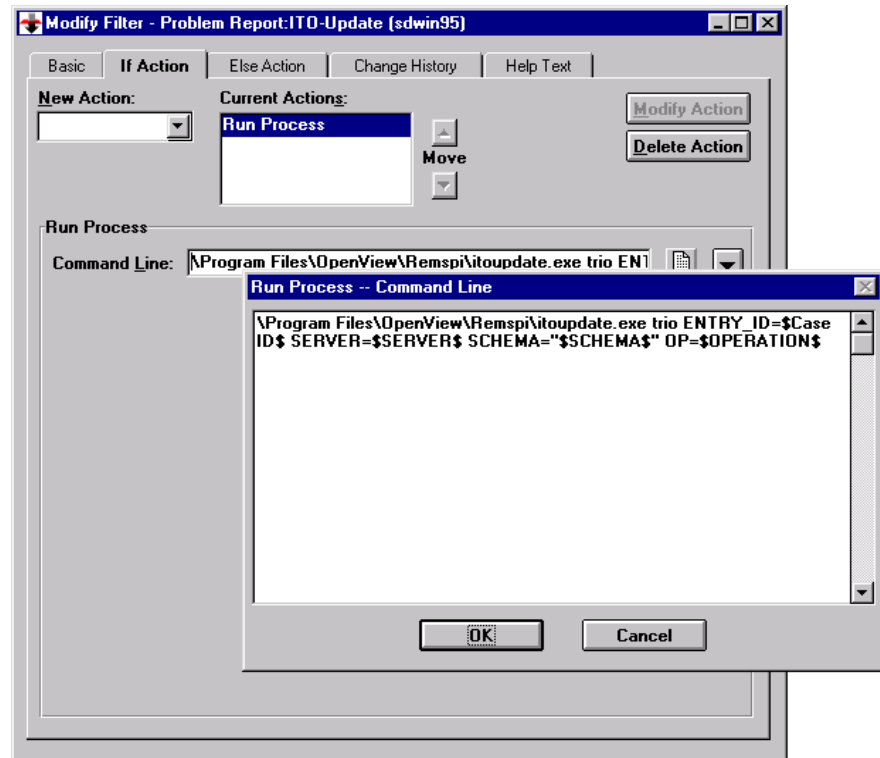
Figure 6-9 Creating a New Filter





2. Add a run-process action that calls the SPI for Remedy ARS client as shown in Figure 6-10.

Figure 6-10 Defining a Filter's Run Process



### Configuring the IF\_MODIFIED ARS Filter manually for itupdate

This special filter is used to notify the SPI for Remedy ARS server that a certain field in an action-request 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.

---

## Customizing the SPI for Remedy ARS ARS Filters and Active Links

You can configure the filter manually in the same way that you create other filters in an action-request 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](#).

---

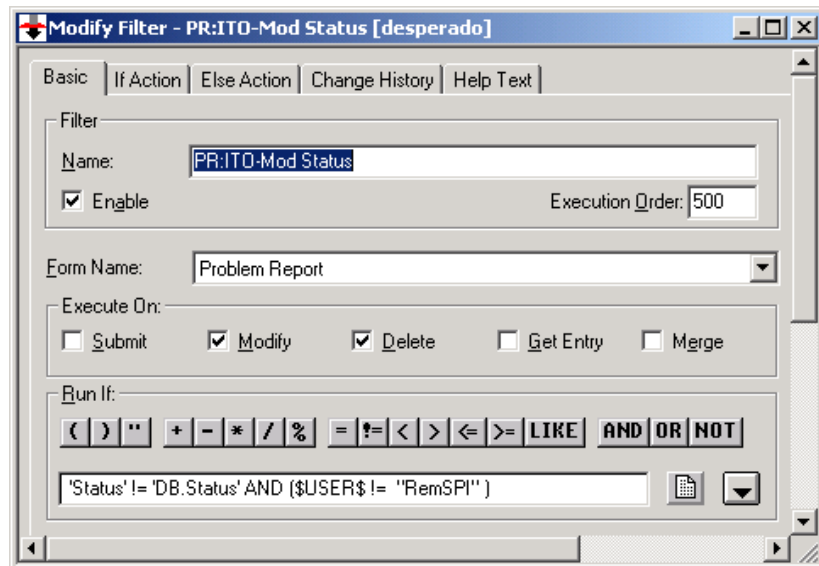
### NOTE

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 SPI for Remedy ARS server (RemSPI in our example). For this filter to be usable, you have to use different accounts than RemSPI for the interactive action-request users.

---

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

**Figure 6-11** Using Special Filters to modify the Status field



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

---

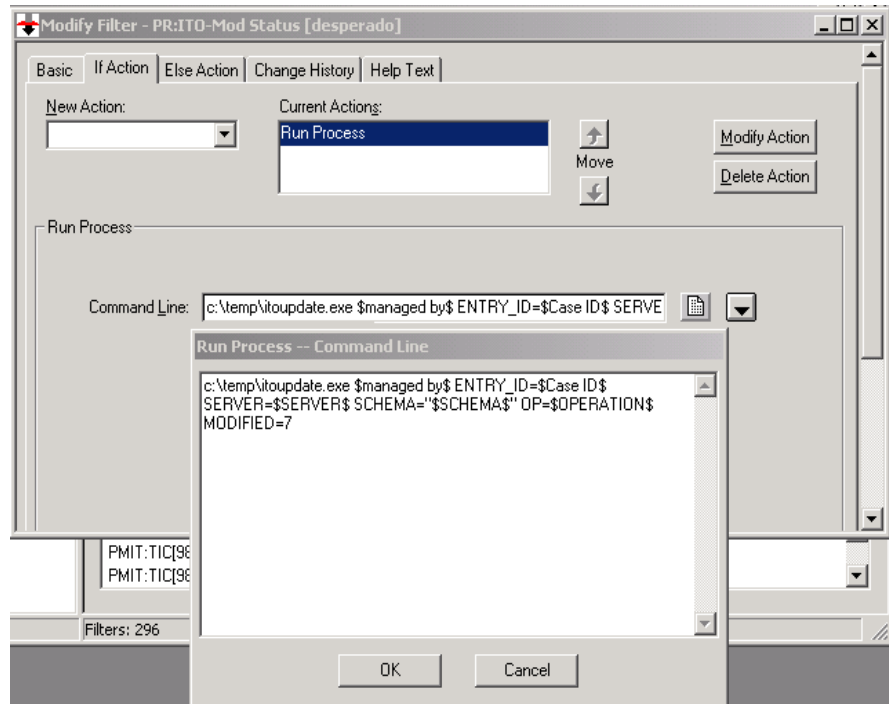
**NOTE**

---

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\itupdate.exe $managed by$ ENTRY_ID=$Case ID$ SERVER=$SERVER$ SCHEMA="$SCHEMA$" OP=$OPERATION$ MODIFIED=7`

**Figure 6-12** The special filter's Run Process



### Configuring an Active Link for itupdate

To configure an active link that enables users to start actions in HPOM:

1. Use the ARS to create the appropriate buttons or menu items in the ARS schema as shown in [Figure 6-13](#).

**Figure 6-13** Buttons for Active Links

The screenshot shows a form with a light green background. At the top, there is a label 'Short Description' followed by a text input field. Below that is a label 'Description' followed by a larger text input field. Underneath is a label 'Work Log' followed by another text input field. At the bottom of the form, there are two buttons: 'Start Action' on the left and 'Escalate' on the right. To the right of each input field, there are small icons representing a calendar and a document.

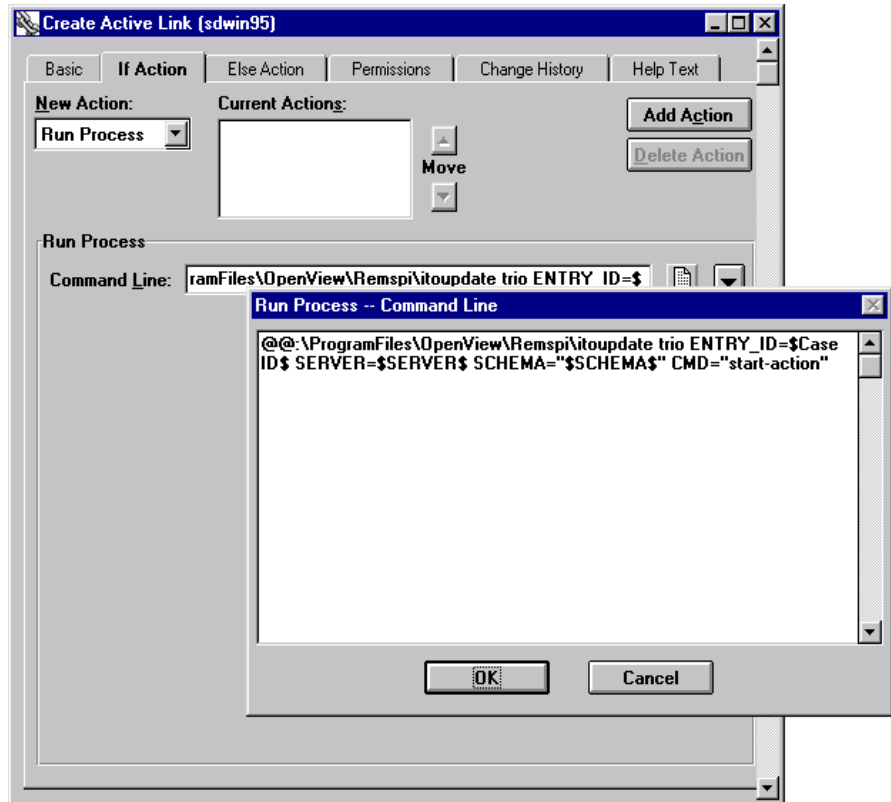
2. Create a new active link that runs when someone uses the menu item or button created in step 1.

**Figure 6-14** A New Active Link

The screenshot shows a dialog box titled 'Create Active Link [sdwin95]'. It has several tabs: 'Basic', 'If Action', 'Else Action', 'Permissions', 'Change History', and 'Help Text'. The 'Basic' tab is selected. The 'Active Link' section contains a 'Name' field with the value 'Problem Report:Action', an 'Enable' checkbox which is checked, and an 'Execution Order' field with the value '500'. The 'Schema Name' is set to 'Problem Report'. The 'Execute On:' section has several checkboxes: 'Submit', 'After Submit', 'Query', 'Return', 'Menu Choice', 'Modify', 'After Modify', 'Set Default', 'Display', 'Window Open', and 'Window Close'. The 'Button/Menu Item' checkbox is checked, and the 'Field' dropdown is set to 'Start Action'. The 'Run If:' section contains a row of logical operators: '(', ')', '&quot;', '+', '-', '\*', '/', '%', '=', '!=', '<', '>', '<=', '>=', 'LIKE', 'AND', 'OR', 'NOT'. Below this is a text input field for a condition.

3. Add a run process that calls the SPI for Remedy ARS client as illustrated in Figure 6-15. 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 86.

Figure 6-15 An Active Link’s Run Process



### Configuring the ARS Filter manually for ItoUpdatePlugin

---

**NOTE**

The configuration of an AR Filter API Plugin requires some knowledge about the AR System Server, especially the configuration of AR Plugins. Thus, the following steps should be performed by or with the help of an AR System Server administrator.

---

Instead of configuring an AR filter action of type *Run Process* to send AR ticket updates back to the SPI for Remedy ARS server by executing the SPI for Remedy ARS client process `itoupdate` on the AR server, it's possible to use a Java Filter API Plugin. This plugin is loaded by the AR Plugin Server at startup and executed via an AR filter action of type *Set Fields*. The use of a plugin is more efficient than starting an external process for every ticket update.

In order to configure the Java Filter API Plugin, perform the following steps:

1. Make sure the AR Plugin Server is installed and started with the AR System Server.
2. Install and register the `ItoUpdatePlugin.jar` plugin on the AR System Server. See [“Method 2: The ItoUpdatePlugin plugin” on page 140](#).
3. The new AR Filter API Plugin needs to send exactly the same information to the SPI for Remedy ARS server as the former SPI for Remedy ARS client `itoupdate` did.

Part of this information is the name of the current AR form/schema, the name of the current AR server and the operation that is performed.

Unfortunately, this information is not available in an AR action of type *Set Fields* using the *Filter API* as its data source. Thus, this information has to be added to the current AR ticket as field values, because fields may be accessed in the aforementioned action.

In the AR form/schema, define 3 new text fields, which will hold the needed information. In the current example, these new fields are `OPERATION (SET)`, `SCHEMA (SET)` and `SERVER (SET)`.

Then, in the AR filter, which is used to send the update information to the SPI for Remedy ARS server via the new plugin, define a *Set Fields* action that stores the needed data in the new fields.

The following screenshot shows the definition of this action.

**Figure 6-16** Store information for the plugin in helper fields

Field	Value
e OPERATION (SET)	\$OPERATION\$
e SCHEMA (SET)	\$SCHEMA\$
e SERVER (SET)	\$SERVER\$

This step is not necessary if your AR form/schema already contains the necessary information as field values, which may then be accessed in the actual *Set Fields* action starting the plugin.

4. The new AR Filter API Plugin needs to be started via a *Set Fields* action. Note that this action must be executed *after* the necessary information has been added to the current ticket with the *Set Fields* action described in the previous step.

The definition of the new action is shown in the next screenshot.

**Figure 6-17** Collect data for ItoUpdatePlugin

**Set Fields**

Data Source: FILTER API

Server Name: arstest1

Plugin Name: HP.AR.FILTER.UPDATEPLUGIN

Position	Value	Type
1	\$managed by\$	Field
2	"ENTRY_ID= "	Char
3	\$Case ID\$	Field
4	"SERVER= "	Char
5	\$SERVER (SET)\$	Field
6	"SCHEMA= "	Char
7	\$SCHEMA (SET)\$	Field
8	"OP= "	Char
9	\$OPERATION (SET)\$	Field

Field	Value
Work Log	\$1\$ + ":" + \$2\$

The *Data Source* is set to *Filter API* and the *Plugin Name* to the name configured in the *ar.cfg* file. The parameters, which are passed to the plugin upon execution, are specified as a table. Note that the data is exactly the same as is described in the *Remedy SPI Administrator's Reference*, where the command line syntax for the SPI for Remedy ARS client *itoupdate* can be found.

The difference is that the parameters for the plugin have to be splitted into a fixed and a variable part.



Example: The 2nd parameter is `ENTRY_ID=${Case ID}`. Because the AR plugin server does not expand variables that are part of a larger string (as shown above), the parameters are splitted into position 2 (fixed part): `ENTRY_ID=` and into position 3 (variable part): `${Case ID}`.

In order to send these split parameters as one parameter to the SPI server, the fixed parameter is terminated with the underscore character `_`. The plugin substitutes a trailing `_` with the value of the next parameter, i.e. the expanded case id of the ticket in this example.

The same applies for the next parameters in position 4/5, 6/7 and 8/9. Note that the `SCHEMA`, `SERVER` and `OPERATION` values are fetched from the newly created ticket fields. The reason is that keywords for this type of information are not available for this data source.

The plugin returns 2 result values, which may be accessed in the *Set Fields* action with variables `$1` and `$2`. These return values may be added to the current ticket, `$1` represents the result code of the transmission (0 stating no error) and `$2` expands to the error text (if any).

In the example shown above, this information is added to the tickets work log.

5. Restart the AR server processes in order to activate the plugin definition.
6. The AR Filter API Plugin of the Remedy SPI uses the logging mechanism of the AR plugin server, i.e. `log4j`. Thus, in order to configure logging, modify the file `log4j_pluginsvr.xml`. The standard log file is `<AR_SYSTEM_HOME>/Arserver/Db/arjavaplugin.log`

---

## Customizing the SPI for Remedy ARS for Multiple Servers

The SPI for Remedy ARS enables you to use more than one action-request server, more than one HPOM 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 HPOM and action-request servers. This section provides information about the following topics:

- [“Using Multiple Action-Request Servers”](#)
- [“Using Multiple HPOM Servers”](#)

### Using Multiple Action-Request Servers

If you want the SPI for Remedy ARS server to create and update action requests on more than one action-request server, create appropriate targets in the rules file. Define different action-request servers using the target’s `SERVER` attribute. For more details on defining targets, see [“Defining Targets” on page 82](#). This section provides information about the following additional topics:

- [“Moving a Ticket Between Action-Request Servers”](#)
- [“Notify the SPI for Remedy ARS Server about the Ticket Move”](#)
- [“Telling the SPI for Remedy ARS How to Handle the Ticket Move”](#)

#### Example 6-4

#### Defining Targets with Multiple Action-Request 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 "C338D5F22121E076C200F6E6B4E94B2C679B2DA"
```

[Example 6-4](#) shows two target definitions from a rules file. The first defines a target on an action-request server in the USA. The second defines a target on an action-request server in Europe.

To specify which action-request 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 104.

### **Moving a Ticket Between Action-Request Servers**

An action-request ticket may be moved from one to an other action-request server, this is known as a DSO transfer and performed as an action-request server function.

---

#### **NOTE**

The actual move of the action-request ticket is done by the action-request server and not by the SPI for Remedy ARS. The SPI is only notified after the action-request server has completed the move to update the SPI databases.

---

In addition to the steps which have to be done in the action-request server configuration (see the action-request 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 for Remedy ARS 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 action-request form.

Here is the syntax how the `itoupdate` program must be called in case an action-request ticket has been moved between action-request 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>
```

### Telling the SPI for Remedy ARS How to Handle the Ticket Move

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

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

In case of a DSO transfer, the new ticket has not been created by the SPI for Remedy ARS, but by the action-request server. Thus, no such information about the *creating* condition is available, but this information is needed for later manual attachments or changes to associated HPOM messages which is expected to result in a modification of the action-request 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 action-request ticket is marked with the keyword `DSO_TARGET`.

If the SPI server receives a move notification, all conditions which have the target action-request 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 action-request ticket and the associated HPOM messages is removed, but no new association is established.

---

In addition, the HPOM message which is associated with the moved action-request ticket may be modified.

In order to modify the HPOM message in the context of the target from which the message is moved away, use the `ON_MOVE_FROM_ME` keyword in the target definition section.

In order to modify the HPOM message in the context of the target to which the message is moved to, use the `ON_MOVE_TO_ME` keyword in the target definition section.

For details, see “Defining Move-From and Move-To Rules” on page 103.

## Using Multiple HPOM Servers

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

If each of your SPI for Remedy ARS servers has exclusive access to a set of action-request schemas, you can customize the SPI for Remedy ARS servers normally. Even if the schemas are on the same action-request 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:

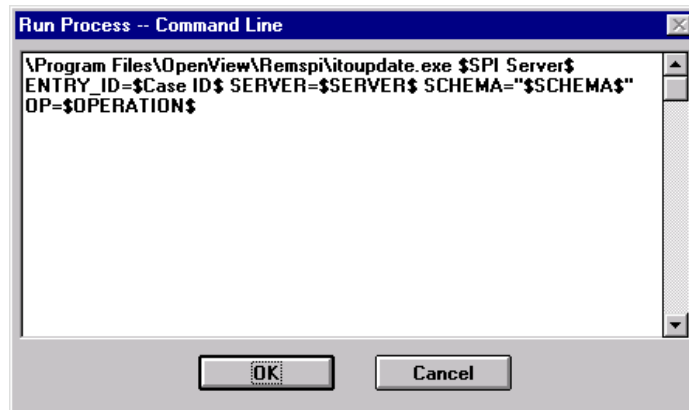
1. Create a field in the schema to store the name of the SPI for Remedy ARS server. This can be a hidden field if the action-request system users do not need to see it.
2. Amend the filter that calls the SPI for Remedy ARS client. Reference the new field as the first parameter in the SPI for Remedy ARS 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.

## Customizing the SPI for Remedy ARS for Multiple Servers

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 4.06”](#) on page 38.

**Figure 6-18** Filter Run-Process for Multiple HPO Servers



- 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 111.

[Example 6-5](#) shows a default action for a target. The action puts the name of the SPI for Remedy ARS server in field 1000000006.

**Example 6-5** Default Action for Multiple HPOM Servers

```
TARGET "ProbSysEUR"
  SERVER "ars1.bbn.hp.com"
  SCHEMA "Problem Report"
  USER "spi_user"
  PASSWORD "C338D5F22121E076C200F6E6B4E94B2C679B2DA"

FIRST FIELD 1000000006 "$HOST_NAME$"
```

## Using Backup Action-Request-System Servers

The SPI for Remedy ARS allows you to specify at most one action-request-system server as a backup server. If the primary action-request-system server is not reachable and the action-request ticket cannot be created, then, if a backup server is defined, the action-request ticket is created on the backup server. Updates to action-request tickets are performed on the action-request-system server where the ticket was originally created.

Note that the backup ARS server needs the same configuration as the primary ARS server, that is: the name of the action-request form (schema), the action-request user (and user password) defined on the backup ARS server must be identical to the values used in the configuration on the primary server.

In the event that neither the primary nor the backup action-request server is reachable, then - if configured - the request is buffered.

The decision to use the backup server is taken each time action-request requests the creation of a ticket.

## Maintaining a backup copy of the SPI database

The SPI for Remedy ARS stores the information regarding the association between HPOM messages and action-request tickets in a separate database. When in use by the SPI for Remedy ARS server, the database is locked to prevent concurrent access. If there is a need to copy this SPI database, the SPI server would need to be stopped to guarantee a consistent copy.

To allow on-line backups of the SPI for Remedy ARS database, an alternative directory for the SPI for Remedy ARS database may be specified in the `remspi.cfg` file. This alternative directory is synchronized by the SPI for Remedy ARS server and always contains a consistent copy as long as the access rules described in this section are obeyed.

[Example 6-6 on page 160](#) shows how to enable the alternate SPI for Remedy ARS database by adding (or activating) the line displayed in the `remspi.cfg` file. You will have to replace the name of the directory used in the example with the name of the directory on your system.

### Example 6-6

#### Configuring SPI DB Backup Copy

```
REMSPI_SPIDB_ALTDIR /mnt/MyAlternateDBDir
```

In addition, the file `remspi_altdb.sync` has to be created as an initially empty file in this directory to enable the alternative SPI for Remedy ARS database. The SPI for Remedy ARS server maintains a request ID in this file to determine when the alternative database needs to be synchronized.

The SPI for Remedy ARS server now keeps a copy of its internal database in this directory. Before writing to this alternative database, the SPI for Remedy ARS server tries to create the file `remspi_altdb.lock` in this directory. If the file can exclusively be created (it does not exist before), then the lock is assumed to be granted and the alternative database is propagated. Note that the lock file is deleted after the write operation is finished.

So, if an external process needs to copy a consistent alternative SPI for Remedy ARS database, the lock file `remspi_altdb.lock` must be exclusively created (must not exist before). After the copying the data, the external process must remove the lock file.

In addition, the external process cannot expect the data to be consistent if the lock file already exists. The existence of the lock file indicates that the SPI for Remedy ARS server is currently propagating the alternative database.



If the alternate directory is not available when the SPI for Remedy ARS server tries to propagate the alternative SPI for Remedy ARS database, a note is written to the SPI for Remedy ARS log file. The directory is considered to be unavailable if the `remspi_altdb.sync` file is not readable.

## Using the SPI for Remedy ARS in Cluster and Load Balancer Environments

If you want to use the SPI for Remedy ARS on nodes in a high-availability cluster or in load balancer environments, you need to take into consideration several important configuration issues. This section provides information about the following topics:

- [“Remedy ARS Server on a Cluster Node”](#)
- [“Two Remedy Servers behind a Load Balancer”](#)
- [“Two Remedy Servers behind a Load Balancer”](#)

---

### 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 about how to set up the HPOM management server or the HPOM managed node on cluster nodes, see the appropriate HPOM documentation.

---

### Remedy ARS Server on a Cluster Node

In this section, the Remedy ARS server running on a cluster node is referred to as the *AR package*. If you are configuring the SPI for Remedy ARS to run in a cluster environment, you will need to consider the following points:

- Monitoring
  - To enable the AR package:

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

```
— opctemplate -e "RemSPI_ARS_server_UX"  
— opctemplate -e "RemSPI_ARS_server_UX_Up"  
— opctemplate -e "RemSPI ARS Server errors (UX) "
```
  - To disable the AR package:

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

## Using the SPI for Remedy ARS in Cluster and Load Balancer Environments

```

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

```

- Communication: HPOM server to Remedy ARS server

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

- Communication: action-request server to HPOM server

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

### HPOM Server on a Cluster Node

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

- Monitoring

— To enable the HPOM package:

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

```
— opctemplate -e "RemSPI Monitor SPI Server"
```

— To disable the HPOM package

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

```
— opctemplate -d "RemSPI Monitor SPI Server"
```

- Communication: Remedy ARS server to HPOM server

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

### Two Remedy Servers behind a Load Balancer

In this scenario, there are two physical Remedy ARS servers running simultaneously, and there is a load balancing system installed. The load balancer has a virtual name and IP address (referred to as *load balancer name/IP* below) which then points to any one of the physical Remedy ARS servers.

If you are configuring the SPI for Remedy ARS to run in a load balancing environment, you will need to consider the following points:

- Monitoring

The policies to monitor the Remedy ARS servers should be assigned to both physical nodes running Remedy ARS servers. These are all policies in these policy groups, depending on the OS of the Remedy ARS server nodes:

- Remedy SPI - AR Server AIX

- Remedy SPI - AR Server NT

- Remedy SPI - AR Server UX

- Communication: HPOM server to Remedy ARS server

In the `rules` file, use the load balancer name/IP as the name of the `TARGET`.

- Communication: Remedy ARS server to HPOM server

The filters and active links on the Remedy ARS server starting the SPI client `itupdate` have to pass the *load balancer name/IP* for the sender ID as a parameter.

## Using External Actions and Data

The first time the SPI for Remedy ARS submits an HPOM message to the action-request system, you can use external data and start external actions. However, we strongly recommend that you use this feature if, *and only if*, you are an experienced shell programmer. This section covers the following areas:

- “Starting External Actions”
- “The `remspiupd` Command”
- “Using External Value Files”
- “Formatting External Value Files”
- “Referencing External Values”

### Starting External Actions

To submit an HPOM message to the SPI for Remedy ARS, the script `remspisub_auto.vbs` (Trouble Ticket interface) passes the message ID to the `remspiupd` command, which forwards the information to the SPI for Remedy ARS server process. The two scripts 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 HPOM 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 HPOM. The command can be used both to submit and update HPOM messages and is called by the `remspisub_*` scripts. For more information on the `remspisub_*` scripts, see “Data Components” on page 29.

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

## Customizing the SPI for Remedy ARS

### Using External Actions and Data

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

where at least one HPOM 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* HPOM message IDs must be specified.
- `-d (ebug)`  
option writes *debugging* information to `stdout`
- `-t (race)`  
option writes *trace* information to `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 HPOM GUI to immediately display the result of the submission. The default submission scripts use the `-q` option for automatic submission by the HPOM server.
- `-u (user)`  
option allows to specify the HPOM user submitting the message

If you are using the `itoupdate` command to submit a message, *no* HPOM messages must already be assigned to the action-request 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* HPOM message must already have been submitted

- The previously submitted HPOM messages must be associated with only *one* action request
- There must be at least *one* HPOM message which has *not* yet been assigned

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 (zero) if the submission succeeds and a non-zero (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 175](#).

### 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 an HPOM messages. These external values are stored in an external file, an example of which is located in:

```
%OvShareDir%\conf\remspi\extval
```

You can use external value files to add information to an action request that is not available in the original HPOM message. An external value file can be either one of the following:

- Static and rarely change
- 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 script `remspisub_auto.vbs`, which submits the HPOM message to the SPI for Remedy ARS server (and thus, to the action-request server). If you need to configure dynamic access to the external values you define, this script 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.

---

**NOTE** If the external value file does not exist, an error message is logged.

---

### 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 6-7](#) shows a simple external value file that contains extra details about a printer.

#### Example 6-7

##### 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 is 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 111.

Example 6-8 shows an action that puts external data from Example 6-7 into a text field. Figure 6-19 shows the result in the action request.

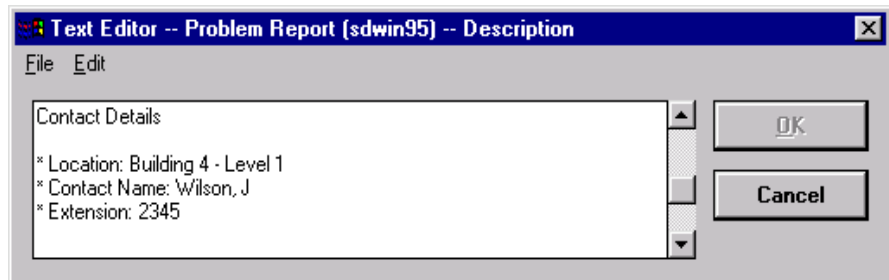
### Example 6-8

#### 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.

---

## Miscellaneous Configurations Tasks

The SPI for Remedy ARS may be configured at a global level by adding special configuration lines to the `remspi.cfg` file. This section describes the following areas:

- [“Setting the Date and Time Format” on page 170](#)

### Setting the Date and Time Format

The format which is used to convert date and time information to text strings may be configured with the following entries in the `remspi.cfg` file:

- `REMSPI_DATE_FORMAT`

This format is used whenever a time value (whether it’s extracted from a HPOM message or an AR ticket field) should be converted to a date string. Default is: `%m/%d%%Y`, e.g. `12/31/2007`

- `REMSPI_TIME_FORMAT`

This format is used whenever a time value (whether it’s extracted from a HPOM message or an AR ticket field) should be converted to a time string. Default is: `%H:%M:%S`, e.g. `11:57:32`

If a time value should be converted to a combined date and time field, both format definitions are used. For details regarding the variables used in these formats, see the documentation of the UNIX `date` command or the `strftime()` API call.

## Utilities installed with the SPI

The SPI for Remedy ARS installs utilities which you may use in your own scripts. This section describes the following utilities:

- “Direct access to AR tickets: `remspi_arutil`” on page 171

### Direct access to AR tickets: `remspi_arutil`

Direct access to existing action-request tickets is available via the utility `remspi_arutil`. With this command, field contents of existing tickets may be retrieved and modified, without using the SPI configuration or the SPI server itself.

Syntax of `remspi_arutil`:

```
remspi_arutil -help
remspi_arutil -version
remspi_arutil -get AR-Spec
remspi_arutil -set AR-Spec -value value
```

The third type of command retrieves the contents of the specified field of an existing ticket, the fourth type of command may be used to set the field to a specified *value*.

The action-request server, the login data and field information is abbreviated as *AR-Spec* above and consist of the following elements:

- `-server` *hostname*

The hostname or IP address of the action-request server on which the ticket exists. This parameter is mandatory.

- `-schema` *formname*

The name of the action-request form in which the ticket exists. This parameter is mandatory.

- `-user` *username*

The name of the action-request account which is used to access the ticket. This parameter is mandatory.

- `-passwd` *password*

### Utilities installed with the SPI

The password of the action-request account specified with `-user`. Note that the password must be specified in its encrypted form, i.e. the output of the SPI tool `remspipasswd` must be used. This parameter is optional and may be omitted if the action-request account is configured without a password.

- `-ar ticket_id`

The ID of the action-request ticket which should be accessed. This parameter is mandatory.

- `-fieldid fieldid`

The numerical id of the action-request ticket field. This parameter is mandatory.

#### Example 6-9

#### Example of `remspi_arutil`

```
remspi_arutil -get -server arstest1 \  
              -schema "Problem Report" \  
              -user Demo \  
              -ar 0000000002901 -fieldid 5
```

```
Problem Report@arstest1[0000000002901]:5 = Demo
```

---

---

**7**

# Troubleshooting the SPI for Remedy ARS

This section describes how to troubleshoot the Smart Plug-in for Remedy Action Request System.

## **In this Section**

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

- “Error Logs and Tracing”
- “Troubleshooting the SPI for Remedy ARS Installation”
- “Troubleshooting SPI for Remedy ARS Rules”
- “Troubleshooting SPI for Remedy ARS Usage”
- “Troubleshooting the SPI for Remedy ARS Internal Database”
- “Troubleshooting the SPI for Remedy ARS Performance”
- “Handling Message Storms”

## Error Logs and Tracing

### Log File Locations

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 175](#) 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**      **Log-File Locations**

File Name	SPI for Remedy ARS System	Location
itoupdate.log	Client	Unix: /var/opt/OV/log/remspi Win: %OvAgentDir%\log\remspi  In case the default directory is not accessible, e.g. if the ARS server is not a managed node, then the directory where the itoupdate command is found is used instead.
remspi.log	Server	%OvDataDir%\log\remspi
remspi.diag		%OvDataDir%\log\remspi
remspiupd.trace		%OvDataDir%\log\remspi
remspi_submit.log		%OvDataDir%\log\remspi
remspi.trc		%OvDataDir%\log\remspi

### Traditional Tracing

You can control the amount of information written to the trace file in the SPI for Remedy ARS configuration file `remspi.cfg`.

The keywords used for tracing are listed in [Table 7-2 on page 176](#).

**Error Logs and Tracing**

The `itupdate` command on the SPI for Remedy ARS client writes error messages to the `itupdate.log`. The `itupdate.log` itself is written to the directory in which the `itupdate` command is installed. The path is determined by the execution of the action-request filter itself. For example, if the action-request filter executes `/abc/itupdate`, then the log file will be written to `/abc/itupdate.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 185](#).

The `remspi.diag` contains diagnostic messages from the SPI for Remedy ARS server. For more information, see [“Troubleshooting the SPI for Remedy ARS Performance” on page 191](#).

The `remspiupd.trace` contains trace information relating to attempts by the SPI for Remedy ARS to submit HPOM messages to the ARS and attach modifications to existing HPOM messages. For more information, see [“The remspiupd Command” on page 165](#). Note that this file is not written by default.

The `remspi_submit.log` contains information relating to the manual or automatic submission of HPOM messages to the ARS. For more information, see [“Troubleshooting SPI for Remedy ARS Usage” on page 185](#).

If the maximum size and number of log files is specified, the SPI keeps a specified number of recent files, adding a suffix `.n` to the filename. Older files get higher numbers (starting with 1), the current file keeps its usual name (without suffix).

The `remspi.trc` file contains information relating to the internal control flow of the SPI programs, especially the internal processing in the SPI server.

**Table 7-2**

**Keywords for Tracing**

<b>Keyword</b>	<b>Parameter</b>
SPI_TRC_LEVEL	An integer value ranging from 1 to 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.



**Table 7-2**                      **Keywords for Tracing (Continued)**

<b>Keyword</b>	<b>Parameter</b>
SPI_DIAG_MODE	The integer 0 (no diagnostic logging) or 1 (diagnostic logging enabled).
SPI_LOG_MAXFILES	Maximum number of log files for the SPI server processes. Default if this keyword is omitted: 1
SPI_LOG_MAXSIZE	Maximum (approx.) size of each SPI log file in kB. Default if this keyword is omitted: 0, which does not impose a limit on the file size.
SPI_DIAG_MAXFILES	Maximum number of diagnostic files for the SPI server processes. Default if this keyword is omitted: 1.
SPI_DIAG_MAXSIZE	Maximum (approx.) size of each SPI diagnostic file in kB. Default if this keyword is omitted: 0, which does not impose a limit on the file size.

To turn tracing on, you have to follow these steps:

- Add/enable the line `SPI_TRC_LEVEL 5` in the `remspi.cfg` file (set a trace level according to your needs)
- Restart the SPI server using the command sequence  
`net stop RemSpiSrv & net start RemSpiSrv`

To turn tracing off, you have to follow these steps:

- Delete/comment the line `SPI_TRC_LEVEL n` in the `remspi.cfg` file
- Restart the SPI server using the following commands:  
`net stop RemSpiSrv & net start RemSpiSrv`

---

**NOTE**

Essential information is written to the trace file if the trace mode is turned on before the SPI for Remedy ARS server is started. Thus, this should be the preferred way to produce a trace file.

---

### Enhanced Tracing

Traditional tracing results in one trace file used by all SPI for Remedy ARS processes and tools. The location and the name of the trace file is predefined and may not be changed. In addition, the trace file grows without bounds.

In situations where a problem occurs rarely, activating the trace mode for a long time may result in a very large trace file.

Enhanced tracing may be used in these situations. Enhanced tracing differs from traditional tracing as follows:

- Separate trace file sets are used for each process and tool
- Number of trace files may be specified for each set
- Maximum size and lifetime for each trace file may be specified
- Location and name for the trace files may be specified

Using enhanced tracing, the filesystem usage for trace information may thus be limited.

Enhanced tracing is configured in the `remspi.cfg` file using the following keywords. Note that traditional tracing has to be activated by setting a trace level greater 0 in order for enhanced tracing to work.

**Table 7-3**

#### Keywords for Enhanced Tracing

Keyword	Parameter
SPI_ETRC_ENABLED	An integer value ranging from 0 to 1. A value of 0 disabled the enhanced trace mode, a value of 1 enabled the enhanced trace mode. Note that enhanced tracing uses the same trace level as traditional tracing. Default: 0

**Table 7-3**

**Keywords for Enhanced Tracing (Continued)**

<b>Keyword</b>	<b>Parameter</b>
SPI_ETRC_PATHNAME	<p>A string specifying the pathname of the trace files used in enhanced tracing.</p> <p>The trace files will be created using this name, with a file extension indicating the file generation. The most current file has generation number 1, older files have increasing generation numbers.</p> <p>Default: Pathname of traditional trace file.</p>
SPI_ETRC_MAXFILES	<p>The maximum number of trace files used in a set, where one set is created for each process and tool. If a limit (size, lifetime) for the currently active trace file is reached, it is closed and all trace files will have their generation number increased by one. The oldest trace file is deleted if its generation number exceeds this parameter. A new active trace file will be created with generation number 1.</p> <p>Default: 1</p>
SPI_ETRC_MAXSIZE	<p>This parameter limits the size of each trace file. The size limit is specified in kB. If a trace file exceeds this size limit, it is closed. See SPI_ETRC_MAXFILES for details.</p> <p>The actual maximum size of a file may be slightly larger than this limit (max. 20kB).</p> <p>Default: 0 (no size limit)</p>
SPI_ETRC_MAXTTL	<p>This parameter limits the lifetime of each trace file. The maximum lifetime is specified in seconds. If a trace file is older than this limit, it is closed. See SPI_ETRC_MAXFILES for details.</p> <p>Default: 0 (no time limit)</p>

### **Tracing the Rules**

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.

- `%OvInstallDir%\bin\remspi\remspisrv -debug`

This keeps the `remspisrv` process in the foreground and prints trace information to `stderr` during the parsing phase of the `rules` file.

## Troubleshooting the SPI for Remedy ARS Installation

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

1. In the HPOM GUI, check that the following elements have been added:
  - Node Groups:
    - Remedy SPI Servers
    - AR Servers NT
    - AR Servers UX
    - AR Servers AIX
  - Tool Groups:
    - Remedy SPI Admin
  - User Roles:
    - remspi\_adm
    - remspi\_op
  - Policy Groups:
    - Remedy SPI - AR Server NT
    - Remedy SPI - AR Server UX
    - Remedy SPI - AR Server AIX
    - Remedy SPI - ITO
2. If either of these steps fails to produce the desired results:
  - a. Check the following log files for more specific information relating to problems concerning the installation of the SPI for Remedy ARS:
    - %OvInstallDir%\install\remspi\log\RemedySPI\_install.log
  - b. Cross check your installation with the information provided in [Table 3-5](#), “File Locations on the HPOM Management Server,” on page 42.

---

**NOTE**

After you install the SPI for Remedy ARS client program on an action-request 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 `%WINDIR%\system32\drivers\etc\services` file on the SPI for Remedy ARS server machine.

---

To modify port a number:

1. On the HPOM management server, open the `services` file in a text editor.
2. Find the section that begins with the following comment:  

```
# HP Operations Manager Remedy SPI entries
```

If no such section exists, add a new section with appropriate comment.
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 action-request server system, too.

---

## Troubleshooting SPI for Remedy ARS Rules

The `rules` file determines how the SPI for Remedy ARS should react to messages from HPOM that are intended for ARS, and action requests from the action-request system, which are intended for HPOM. For more information about the syntax and structure of the `rules` file, see “Customizing the Rules File” on page 76.

---

### 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' ...
Connection 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 action-request 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 action-request system error number (in this example; 314). For more information about the error, use the error number, for example 314, to refer to the action-request-system documentation.

---

**NOTE**

If the `remspiconfig` command starts while the SPI for Remedy ARS server is running, the SPI for Remedy ARS displays an error message indicating that some files could not be locked. The problem occur because both the SPI for Remedy ARS server and the `remspiconfig` command try to lock the SPI for Remedy ARS databases.

---



## **Troubleshooting SPI for Remedy ARS Usage**

This section describes some of the problems that can occur when using the Smart Plug-in for Remedy Action Request System. Some of the problems are indicated by the arrival of a message in the HPOM for UNIX message browser. The section covers the following areas:

- [“Name Resolution”](#)
- [“Removing Buffered Action Requests”](#)
- [“Setting the Language of the SPI server”](#)
- [“Duplicated annotations”](#)
- [“Missing annotations for ITO\\_UPDATE actions”](#)
- [“Remedy AR server installed in non standard directory”](#)

### **Name Resolution**

- Symptom

Problems appear to occur during the resolution of IP addresses on Remedy ARS servers with multiple host names (multi-homed hosts).

- Explanation

You can use short host names, long host names, or the IP address of the action-request system when defining names for Remedy ARS servers within the SPI for Remedy ARS. However, the SPI for Remedy ARS uses the following, specific order when resolving these names for Remedy ARS servers:

1. Short hostname
2. Long hostname
3. IP address

For more information on defining an action-request server in SPI for Remedy ARS target definitions, see [“Defining Targets” on page 82](#).

### **Removing Buffered Action Requests**

- Symptom

If you want to clear the list of buffered action requests (if `OPTION BUFFER_TT` is specified), then you may use the tool `remspi_qutil`.

- Explanation

The tool `remspi_qutil` may be called using the following syntax:

**Example 7-2 Usage of `remspi_qutil`**

```
remspi_qutil [-prio HIGH | MEDIUM | LOW] [-mce] -list  
remspi_qutil [-prio HIGH | MEDIUM | LOW] [-mce] -delete ARS
```

Where:

- `prio`  
specifies the priority of the queue. Default: LOW
- `mce`  
uses the queued message change events instead of the queued submit requests
- `list`  
lists the entries of the specified queue
- `delete`  
deletes entries of the specified queue which are targeted to the named ARS server

### Setting the Language of the SPI server

- Symptom

If you experience problems creating new tickets on an action-request server, and the server rejects the requests with complaints about illegal user names or passwords, it may be that the SPI for Remedy ARS server is running in a language environment that is not supported by the action-request server. You can check this by enabling trace level 6 and inspecting the return parameters of the action-request API calls.

- Explanation

To start the SPI for Remedy ARS server in a specific language environment, you may configure the language in the `remspi.cfg` file.

To start the SPI for Remedy ARS 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 HPOM messages or duplicated entries in action-request tickets, you may decide to reconfigure your rules file or to establish a new action-request user performing the action-request ticket updates.

- Explanation

If the rules file is configured in such a way that changes to action-request tickets are added as annotations to the HPOM messages and changes to the HPOM messages are treated as an update request for the action-request 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 action-request ticket was updated manually or by the SPI for Remedy ARS itself.

To help the SPI for Remedy ARS to distinguish between manual and automatic (SPI-initiated) updates, you can use a special action-request user for all SPI for Remedy ARS transactions. This action-request account is then not used for interactive users. See [“Configuring the IF\\_MODIFIED ARS Filter manually for itupdate” on page 145](#) for details on how to configure.

## Missing annotations for ITO\_UPDATE actions

- Symptom

In previous versions of the SPI for Remedy ARS (earlier than 4), the modification of HPOM messages, e.g. via `ITO_UPDATE` actions, triggered an internal HPOM message annotation. In previous versions of HPOM, these annotations could not be avoided. In the newer versions of HPOM, which are supported by SPI for Remedy ARS 4 and higher, it’s possible to suppress such internal annotations.

Starting with SPI for Remedy ARS 4, no internal annotations are created anymore if messages are modified in the `ITO_UPDATE` actions.

- Explanation

Because SPI for Remedy ARS 4 introduced the option `SMART_ANNOUNCE_MCE`, it became necessary to avoid internal HPOM annotations.

The internal HPOM annotations were never documented in the SPI for Remedy ARS and thus were suppressed without further notice.

### **Remedy AR server installed in non standard directory**

- Symptom

If it's not possible to start and stop the AR server using the tools provided by the SPI for Remedy ARS or if messages show up in the message browser indicating that the AR server is not running although you know it is, it may happen that the AR server is installed in a directory unknown to the SPI.

- Explanation

The SPI for Remedy ARS looks in standard places for installed AR servers (`/opt/ar`, `/usr/ar`). If the server is installed in a different directory, it may not be found. In this case, create a symbolic link in `/opt/ar` or in `/usr/ar` which points to the actual installation directory.

## Troubleshooting the SPI for Remedy ARS Internal Database

The SPI for Remedy ARS stores the association between HPOM messages and action-request tickets in its own database. This database can be maintained and verified using the `remspidbutil` command.

### Example 7-3

#### Using the `remspidbutil` Command

```
remspidbutil -help |
             -list [ -target target ... ] |
             -verify [ -target target ... ] |
             -delete [ -target target ... ] [-ar] ID ...
             -modify [ -cleanup ] -old_ovo_id ID -new_ovo_id ID
```

where

- `help`  
Shows the usage information for the tool
- `list`  
Check and list entries in the internal SPI database, but do not connect to action-request servers. If targets from the `rules` file are specified, only entries for those targets are checked and listed, else all targets defined in the `rules` file are processed.
- `verify`  
Verifies the internal database and connect to action-request servers. If a target (see the `rules` file) is specified, then only the database entries for this target are checked, else all targets defined in the `rules` file are checked. If inconsistencies are found, the tool asks for confirmation to cleanup.
- `modify`  
Scans the complete internal SPI database and changes all occurrences of the specified old HPOM message id to the specified new HPOM message id. This may be used if a HPOM message was exchanged by a newer one and the SPI should connect the existing action-request tickets for this old message to the new message. If the option `-cleanup` is specified, inconsistent entries in the internal SPI database are deleted, otherwise a warning is issued. Note that modifications are applied to all targets defined in the `rules` file.

- `delete`

Deletes entries from the internal database. If the option `-ar` is specified, then the action-request ticket ids are expected as *ID ...* arguments, else HPOM message ids are expected. If explicit targets are given, the entries are deleted only when occurring in these associations.

---

**NOTE**

If the `remspidbutil` 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 `remspidbutil` command try to lock the SPI databases.

---

## Troubleshooting the SPI for Remedy ARS Performance

The SPI for Remedy ARS logs diagnostic information regarding incoming events and their processing including timing information to a separate log file. To enable diagnostic logging, add the line `SPI_DIAG_MODE 1` to the `remspi.cfg` configuration file and restart the SPI for Remedy ARS server processes (see “[Error Logs and Tracing](#)” on page 175). If you enable diagnostic logging, the SPI for Remedy ARS logs each event and its processing in one separate text line in the file `remspi.diag` as illustrated in [Example 7-4 on page 191](#).

### Example 7-4

#### The `remspi.diag` File Syntax

```
Date/Time SPI(remspisrv-pid): Init Type Chan Src-Object ->
Tgt-Object MCE-Type (Duration [ms]) (SPI215-msgnum)
```

where

- *Date/Time*  
shows the date and the time of the event
- *pid*  
is the PID of the SPI server process
- *Init*  
is the initiator of the event: OVO or ARS
- *Type*  
is the type of the event: SUB for submitting a HPOM message, MCE for modifying a HPOM message or UPD for update of an AR ticket
- *Chan*  
is the channel via which the event arrived: COM for network communication, INT for events generated internally in the SPI
- *Src-Object*  
is the description of the source object which triggered the event; HPOM messages are listed as `MSG=Msg-ID`, AR tickets are listed as `TTN=Ticket-ID@AR-Form@AR-Server`
- *Tgt-Object*

is the description of the target object which is modified by the event; HPOM messages are listed as MSG=Msg-ID, AR tickets are listed as TTN=Ticket-ID@AR-Form@AR-Server

- *MCE-Type*

is the list of message change events which triggered the event (*Type* is MCE)

- *Duration*

is the duration in milliseconds for processing the event



## Handling Message Storms

The SPI for Remedy ARS may be configured to temporarily skip the creation of action-request tickets in case of a message storm.

If the file `%OvShareDir%\conf\remspi\no_auto_tickets` exists, the script `remspisub_auto.vbs` suppresses the current request and does not forward the message to the SPI server. A note is written to the SPIs log file instead.

There is no need to restart the SPI server when the file is created or deleted. Manually submitted tickets are still processed.



---

---

# 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 Smart Plug-in for Remedy Action Request System software from the systems on which you have installed it. The section covers the following areas:

- “Removing Action-Request Filters”
- “Removing the SPI for Remedy ARS from the HPOM GUI”
- “Removing the SPI for Remedy ARS from the HPOM Server”
- “Removing the SPI for Remedy ARS from the Remedy ARS Server”

## Removing Action-Request Filters

To remove or unregister action-request-system filters, use the `remspifilter` command with the `-d(DELETE)` 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 143](#).

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 action-request server.

---

## Removing the SPI for Remedy ARS from the HPOM GUI

The SPI for Remedy ARS integration with the HPOM GUI has to be removed manually, at least some elements of the integration. You will have to remove the following components from the HPOM GUI:

- Managed Nodes Groups

Note that you should not remove the action-request servers from the managed nodes repository until you have removed the HPOM agent software including the SPI for Remedy ARS policies. For more information, see [“Removing the SPI for Remedy ARS from the Remedy ARS Server”](#) on page 200.

- SPI Tool Group
- SPI Policies

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

## Removing the SPI for Remedy ARS from the HPOM Server

To remove the Smart Plug-in for Remedy Action Request System components from the HPOM management server and complete the general clean up process:

1. On the HPOM management server, stop the SPI for Remedy ARS service. You may use the Services Control Panel from Windows to locate and stop the service or you may execute the command “net stop RemSpiSrv”.

Then use the Windows Add/Remove Software Icon in the Windows Control Panel, select the entry for the HP Operations Manager Smart Plug-Ins and press the Change button. Answer the questions and finally mark the Remedy SPI check box before starting the removal process.

2. Remove the HPOM management server and the action-request servers from the SPI for Remedy ARS node groups. Then re-deploy the policies to the managed nodes using the normal HPOM deployment tools.
3. Cleanup the GUI by removing the SPI for Remedy ARS-specific elements. For more information, see [“Removing the SPI for Remedy ARS from the HPOM GUT” on page 198](#).

## Removing the SPI for Remedy ARS from the Remedy ARS Server

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

To remove the SPI for Remedy ARS instrumentation from the action-request server, re-deploy the instrumentation to the action-request server using HPOM means. On HPOM 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 Remedy ARS servers, you can remove the Remedy ARS servers from the managed-nodes bank in the HPOM GUI.



**Symbols**

\$MSG\_ANNOS\$, 80

**A**

Accessing AR tickets, 171

ACKNOWLEDGE keyword, 87, 97

acknowledging messages, 92, 100

action request, 19

    composite formats, 119

    defining values, 119

    setting default values, 121

    setting update policies, 123

actions

    automatic attachment, 125

    defining, 111

    defining values, 119

    external, 30

    setting default values, 121

    setting enumerated field values, 122

    setting update policies, 123

    starting operator-initiated, 95

    target default, 124

    variables, 113

active links in ARS, 21

adding an annotation, 103

adding message annotations, 93

additional AR user account, 84

ADDITIONAL USER, 84, 112

administrator tools, 64

    remedy SPI status, 64

    SPI for Remedy ARS test, 65

    start AR server, 65

    start SPI for Remedy ARS server, 64

    stop AR server, 65

    stop SPI for Remedy ARS server, 64

ANNO\_FAILED\_MSG\_UPDATE, 79

ANNO\_SEPARATOR, 80

ANNOTATE keyword, 88

annotations

    adding to messages, 93

    duplicated, 187

    troubleshooting, 187

API version, 35

application groups

    Remedy SPI Admin, 40

AR Plugin Server, 150

AR Server AIX policy group, 40, 181

AR Server NT policy group, 40, 41, 181

AR Server UX policy group, 40, 41, 181

AR servers

    SPI for Remedy ARS installation pre-requisites,  
    35

AR Servers AIX node group, 40, 181

AR Servers NT node group, 40, 181

AR Servers UX node group, 40, 181

AR tickets

    direct access, 171

AR\_SESSION\_TIMEOUT\_LONG, 83

AR\_SESSION\_TIMEOUT\_NORMAL, 83

AR\_SESSION\_TIMEOUT\_XLONG, 84

ar.cfg, 152

architecture

    Remedy SPI

        data components, 29

    SPI for Remedy ARS, 27

        components, 27

        data components, 29

        data components config files, 31

        data components database, 31

        data components external actions, 30

        data components rules file, 31

        HPOM integration, 32

        Network, 27

        Processes, 27

ARRPC, 56, 83

ARS

    action request, 19

    active links, 21

    concepts, 19

    filters, 21

    integration with HPOM, 15

    schema, 20

ARS API version, 35

ARS field data types, 112

ARTCPPORT, 56, 83

AS USER, 112

attach rules

    defining, 97, 103

ATTACH\_TO, 125

ATTACH\_TO\_EXISTING, 125, 126

attaching messages, 123

    automatically, 125

attributes

    condition CONDITION, 51

    condition DESCRIPTION, 51

    condition SET TARGET, 51

    target ADDITIONAL USER, 84

    target AR\_SESSION\_TIMEOUT\_LONG, 83

    target AR\_SESSION\_TIMEOUT\_NORMAL, 83

    target AR\_SESSION\_TIMEOUT\_XLONG, 84

    target ARRPC, 56, 83

---

## Index

target ARTCPPOINT, 56, 83  
target BACKUP\_SERVER, 82  
target PASSWORD, 51, 83  
target SCHEMA, 51, 82  
target SERVER, 51, 82  
target USER, 51, 82

### B

BACKUP\_SERVER target attribute, 82  
buffer rules  
  defining, 85  
BUFFER\_TT, 79  
buffering, 79, 81

### C

checking the rules file, 183  
CMA keyword, 92, 99  
command  
  itupdate, 53, 55, 138  
  rempiconfig, 183  
  rempifilter, 55, 141, 143  
  remspiupd, 165  
communication port, 56  
components  
  data, 29  
    config files, 31  
    database, 31  
    external actions, 30  
    rules file, 31  
composite formats  
  to define values, 119  
concepts  
  ARS, 19  
CONDITION condition attribute, 51  
conditions  
  CONDITION attributes, 51  
  DESCRIPTION attribute, 51  
  SET TARGET attribute, 51  
config files for data components, 31  
connection information, 138

### D

data components, 29  
  config file, 31  
  database, 31  
  external actions, 30  
  rules file, 31  
data types, supported, 112  
database

  for SPI for Remedy ARS, 31  
database for SPI for Remedy ARS data components,  
  31  
default  
  values for action requests, 121  
defining  
  action variables, 113  
  actions, 111  
  global options, 78  
  target schema, 82  
  values using composite formats, 119  
defining attach rules, 97, 103  
defining buffer rules, 85  
defining submit rules, 97, 103  
defining update rules, 86  
DESCRIPTION condition attribute, 51  
disable trace, 177  
disk space  
  installation  
    AR server, 37  
DISOWN keyword, 91, 98, 99  
disowning messages, 95, 101  
DSO, 77, 86  
DSO move, 155  
DSO\_TARGET, 156  
DSO\_TRANSFER, 155  
duplicated annotations, 187

### E

enable trace, 177  
enumerated field values, 122  
external actions, 30  
External Value File, size limit of key, 168  
External Value File, size limit of value, 168

### F

field  
  data types, supported, 112  
  enumerated values, 122  
file  
  rules, 127  
file locations  
  binaries, 42  
  configuration, 42  
  documentation, 42  
  HPOM integration, 42  
  HPOM managed node  
    UNIX, 42  
    Windows, 42  
  HPOM management server, 42

- logs, 42
- run-time, 42
- SPI for Remedy ARS client
  - UNIX, 42
  - Windows, 42
- SPI for Remedy ARS server, 42
- temporary, 42
- files
  - configuration for data components, 31
  - rules for data components, 31
- Filter API Plugin, 150
- filters
  - in ARS, 21
  - registering in HPOM, 55
  - remove, 197
  - SPI for Remedy ARS, 141
- Firewall, 56
- flexible ticket updates, 80
- FLEXIBLE\_TICKET\_UPDATES, 80
- FORCE\_OWN keyword, 90, 98
- FORGET\_ACKED\_MSGS, 79
- G**
  - generating an HPOM message, 103
  - global option
    - defining, 78
  - global options
    - ANNO\_FAILED\_MSG\_UPDATE, 79
    - ANNO\_SEPARATOR, 80
    - BUFFER\_TT, 79, 81
    - FLEXIBLE\_TICKET\_UPDATES, 80
    - FORGET\_ACKED\_MSGS, 79
    - NO\_AR\_UPDATE, 78
  - groups
    - node, 61
- H**
  - HPOM
    - ARS integration
      - architecture of SPI for Remedy ARS, 32
      - integration with ARS, 15
    - HPOM managed node
      - UNIX file locations, 42
      - Windows file locations, 42
    - HPOM management server
      - file locations, 42
      - SPI for Remedy ARS installation pre-requisites, 35
  - HPOM users, 62

- I**
  - IF\_MODIFIED, 55, 87, 88, 89, 90, 91, 92, 93, 94, 140, 142, 145, 187
  - info, 138
  - install
    - distributing SPI for Remedy ARS policies, 41
    - nodes and node groups, 41
    - RAM, 37
    - SPI for Remedy ARS software, 38, 40
  - installation
    - pre-requisites, 35
    - AR servers, 35
  - integration
    - HPOM and ARS, 15
    - HPOM and Remedy SPI, 32
  - IPv6, 36
  - itupdate
    - buffering, 138
    - registering ARS filters in HPOM, 55
  - itupdate command, 53, 55, 138
  - itupdate.log, 176
  - itupdate.log file, 175
  - ItoUpdatePlugin, 140, 149
  - ItoUpdatePlugin.jar, 140, 150
- K**
  - keywords
    - ACKNOWLEDGE, 87, 97
    - ANNOTATE, 88
    - CMA, 92, 99
    - DISOWN, 91, 98, 99
    - FORCE\_OWN, 90, 98
    - MSGTXT, 91
    - OP\_ACTION, 89
    - OWN, 89, 98
    - SEVERITY, 92
    - UNACKNOWLEDGE, 88
- L**
  - language, 186
  - LIKE, 104, 110
  - links
    - active in ARS, 21
  - load balancer, 163
  - load balancing environment, 164
  - location of files
    - binaries, 42
    - configuration, 42
    - documentation, 42
    - HPOM integration, 42

---

## Index

- logs, 42
  - run-time, 42
  - temporary, 42
- log file
  - itupdate.log, 175
  - remspi\_submit.log, 175
  - remspi.log, 175
  - remspiupd.log, 165
- logfile
  - configuring trace, 175
  - enable tracing, 177
  - itupdate.log, 176
  - remspi\_submit.log, 176
  - remspi.log, 176
  - remspiupd.trace, 176
- M**
- management server
  - file locations
    - binaries, 42
    - configuration, 42
    - documentation, 42
    - HPOM integration, 42
    - logs, 42
    - run-time, 42
    - temporary, 42
- maxlogfiles, 138
- maxlogsize, 139
- message
  - acknowledging, 92, 100
  - adding an annotation, 103
  - adding annotations, 93
  - generating HPOM messages, 103
  - modifying the text, 96, 102
  - removing ownership of, 95, 101
  - setting CMAs, 102
  - taking ownership of, 95, 101
  - unacknowledging, 93
- modifying message text, 96, 102
- modifying policies, 71
- moving AR tickets, 155
- MSG\_EVT\_ANNO, 120
- MSG\_EVT\_ANNO\_ADD, 107, 121
- MSG\_EVT\_ANNO\_DEL, 107, 121
- MSG\_EVT\_ANNO\_SET, 107, 121
- MSG\_EVT\_CHGSEV, 107, 120
- MSG\_EVT\_CMA\_UPDATE, 107, 120
- MSG\_EVT\_CREATE, 120
- MSG\_EVT\_DISOWN, 120
- MSG\_EVT\_DUPL\_RECEIVED, 107, 120

- MSG\_EVT\_MODIFY, 107, 120
- MSG\_EVT\_MSGCHG, 107, 120
- MSG\_EVT\_OA\_END, 120
- MSG\_EVT\_OA\_START, 120
- MSG\_EVT\_OWN, 120
- MSG\_EVT\_UNACK, 120
- MSGTXTkeyword, 91

## N

- name resolution
  - troubleshooting, 185
- NO\_AR\_UPDATE, 78
- no\_auto\_tickets, 193
- node groups, 61
  - AR Servers AIX, 40, 181
  - AR Servers NT, 40, 181
  - AR Servers UX, 40, 181
  - SPI for Remedy ARS Servers, 40, 181
- nodes
  - assign to SPI for Remedy ARS node groups, 41
- NOT, 104

## O

- ON\_MOVE\_FROM\_ME, 103, 156
- ON\_MOVE\_TO\_ME, 103, 157
- OP\_ACTION keyword, 89
- operations on AR tickets, 84
- operator tools
  - Submit Message, 66
- operator-initiated actions, 95
- OWN keyword, 89, 98
- owning messages, 95, 101

## P

- PASSWORD target attribute, 51, 83
- Plugin, 137, 140, 149
- policies
  - distribute Remedy SPI, 41
  - modifying, 71
  - remove from AR server, 200
  - update for action requests, 123
- policy group
  - Remedy SPI - AR Server NT, 40, 41, 181
  - Remedy SPI - AR Server UX, 40, 41, 181
  - Remedy SPI - ITO, 40, 41, 181
- Private Queue, 56
- ProblemReport.def file, 48

## R

- RAM requirements

- AR server, 37
- HPOM server, 37
- register ARS filters in HPOM, 55
- regular expression, 104, 110
- Remedy API version, 35
- Remedy SPI
  - AR Server AIX policy group, 40, 181
  - AR Server NT policy group, 40, 41, 181
  - AR Server UX policy group, 40, 41, 181
  - architecture, 27
    - components, 27
    - data components, 29
    - HPOM integration, 32
    - processes, 27
  - HPOM users, 62
  - installation pre-requisites, 35
    - AR servers, 35
  - Internal Database
    - troubleshooting, 189
  - ITO policy group, 40, 41, 181
  - Message Storms
    - troubleshooting, 193
  - name resolution
    - troubleshooting, 185
  - node groups, 61
  - nodes and node groups, 41
  - Performance
    - troubleshooting, 191
  - policies
    - distribute, 41
  - removing buffered AR requests
    - troubleshooting, 185
  - remspifilter command, 141
  - rules
    - troubleshooting, 183
  - troubleshooting, 173
    - error logs, 175
    - installation, 181
    - Internal Database, 189
    - Message Storms, 193
    - name resolution, 185
    - Performance, 191
    - removing buffered AR requests, 185
    - rules, 183
    - tracing, 175
    - usage, 185
  - usage
    - troubleshooting, 185
- Remedy SPI Admin application group, 40
- Remedy SPI Admin tool group, 181
- Remedy SPI client
  - file locations
    - UNIX, 42
    - Windows, 42
- Remedy SPI server
  - file locations, 42
- Remedy SPI Servers node group, 40, 181
- Remedy SPI software
  - install, 40
  - installation pre-requisites, 35
    - AR servers, 35
  - remove, 197
  - remove policies, 200
  - remove server, 198, 199
- remove
  - AR filters, 197
  - ARS server
    - policies, 200
  - Remedy SPI server, 198, 199
  - Remedy SPI software, 197
- removing buffered AR requests
  - remspi\_qutil, 185
  - troubleshooting, 185
- REMSPI\_ARRPC, 56
- remspi\_arutil, 171
- REMSPI\_LANG, 186
- REMSPI\_QUEUE\_READ\_HIGH, 85
- REMSPI\_QUEUE\_READ\_LOW, 85
- REMSPI\_QUEUE\_READ\_MEDIUM, 85
- remspi\_qutil, 186
- remspi\_submit.log, 176
- remspi\_submit.log file, 175
- remspi.cfg
  - ARTCPPORT, 56
  - REMSPI\_ARRPC, 56
  - REMSPI\_QUEUE\_CHECK, 81
- remspi.log, 176
- remspi.log file, 175
- remspi.trc, 176
- remspi-adm User, 40, 181
- remspiconfig command, 183
- remspifilter comand, 55, 141, 143
- remspi-op User, 40, 181
- remspisub\_auto.vbs script, 30
- remspiupd command, 165
- remspiupd.log file, 165
- remspiupd.trace, 176
- remspiupd.trace file, 175
- rpc number, 56
- rules
  - checking, 183

---

# Index

- defining attachments, 97, 103
- defining buffer priority, 85
- defining global options, 78
- defining submittals, 97, 103
- defining target schemas, 82
- defining updates, 86
- troubleshooting, 183
- rules file, 50
  - example, 127
- rules file for data components, 31
- Run Process, 137, 150
  
- S**
- schema example, 20
- SCHEMA target attribute, 51, 82
- scripts
  - remspisub\_auto.vbs, 30
- separating multiple annotations, 80
- server
  - Remedy SPI
    - file locations, 42
- SERVER target attribute, 51, 82
- Set Fields, 150
- SET TARGET condition attribute, 51
- setting CMAs, 102
- SEVERITY keyword, 92
- SPI client
  - buffering, 138
- SPI DB maintenance, 79
- SPI\_DIAG\_MAXFILES, 177
- SPI\_DIAG\_MAXSIZE, 177
- SPI\_DIAG\_MODE, 177
- SPI\_ETRC\_ENABLED, 178
- SPI\_ETRC\_MAXFILES, 179
- SPI\_ETRC\_MAXSIZE, 179
- SPI\_ETRC\_MAXTTL, 179
- SPI\_LOG\_MAXFILES, 177
- SPI\_LOG\_MAXSIZE, 177
- SPI\_TRC\_LEVEL, 176
- starting operator-initiated actions, 95
- submit rules
  - defining, 97, 103
  - keyword definitions
    - ACKNOWLEDGE, 97
    - CMA, 99
    - DISOWN, 98, 99
    - FORCE OWN, 98
    - OWN, 98

- T**
- target
  - attributes BACKUP\_SERVER, 82
  - attributes PASSWORD, 51, 83
  - attributes SCHEMA, 51, 82
  - attributes SERVER, 51, 82
  - attributes USER, 51, 82
  - defining, 82
- target default action, 124
- target schema, 82
- Timeout, 83, 84
- tool groups
  - Remedy SPI Admin, 181
- tools
  - administrator, 64
    - SPI for Remedy ARS status, 64
    - SPI for Remedy ARS test, 65
    - start AR server, 65
    - start SPI for Remedy ARS server, 64
    - stop AR server, 65
    - stop SPI for Remedy ARS server, 64
  - operator
    - Manually Submit Message, 66
- trace disable, 177
- trace enable, 177
- trace file
  - remspiupd.trace, 175
- troubleshooting
  - AR server in non-standard directory, 188
  - duplicated annotations, 187
  - Enhanced Tracing, 178
  - error logs, 175
  - installation, 181
  - language, 186
  - Message Storms, 193
  - name resolution, 185
  - Performance, 191
  - Remedy SPI Internal Database, 189
  - Remedy SPI rules, 183
  - Remedy SPI usage, 185
  - removing buffered AR requests, 185
  - the Remedy SPI, 173
  - tracing, 175
  - Tracing the Rules, 180
  - Traditional Tracing, 175

- U**
- UNACKNOWLEDGE keyword, 88
- unacknowledging messages, 93

**UNIX**

- file locations
  - Remedy SPI client, 42
- unregister AR filters, 197
- update policies
  - attached messages only, 123
  - for action requests, 123
- update rules
  - defining, 86
  - keyword definitions
    - ACKNOWLEDGE, 87
    - ANNOTATE, 88
    - CMA, 92
    - DISOWN, 91
    - FORCE OWN, 90
    - MSGTXT, 91
    - OP\_ACTION, 89
    - OWN, 89
    - SEVERITY, 92
    - UNACKNOWLEDGE, 88

**User**

- remspi-adm, 40, 181
  - remspi-op, 40, 181
  - USER target attribute, 51, 82
- Utilities**
- remspi\_arutil, 171

**V****values**

- default for action requests, 121
  - defining using composite formats, 119
  - enumerated field, 122
- variables**
- action definition, 113

**W****Windows**

- file locations
  - Remedy SPI client, 42
  - SPI for Remedy ARS client, 42