Action Request System ™ Accessories

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

SunSoft Part No: 875-1779-10
Revision A, January 1996

Please
Recycle

Adobe PostScript

Printed in the U.S.A.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause in DFAR 52.227-7013 or the equivalent clause in FAR 52.227-19, whichever is applicable.

Cover design by Carlick Advertising.

Part Number: APP-210-DN9-001

Contents

Part 1 —Integration Options with HP OpenView

Part 2 —Integration Options with IBM NetView/6000

Part 3 —Integration Options with SunNet Manager

Preface

The *Action Request System Accessories* manual provides a collection place for documentation that describes special integrations with the Action Request System. As shipped, the book includes modules documenting integration options for three popular network management platforms: HP OpenView, IBM NetView / 6000, and SunNet Manager.

Audience

Each module that becomes a part of the *Action Request System Accessories* manual will provide documentation for users of the Action Request System who also want to understand the features and functions of the accessory or application being described.

The network management integration modules shipped with this manual are intended for anyone who wants to integrate the AR System or its Help Desk Templates software with one of several network management products available from other vendors.

You need to be familiar with the AR System tools, as well as the network management software with which you are integrating the AR System. For information on the AR System or Help Desk Templates, refer to the publications listed in the AR System Documents section on the following page. For information about a specific network management product, refer to the vendor’s documentation.

Network Management Integration Options

The network management platform integration modules are organized alphabetically by the network management product. The following modules are included in this manual.

- ***Integration Options with HP OpenView*** describes how to start the AR System from HP OpenView and how to register the AR System with the HP OpenView event stream.
- ***Integration Options with IBM NetView/6000*** describes how to start the AR System from IBM NetView and how to register the AR System with the IBM NetView event stream.
- ***Integration Options with SunNet Manager*** describes how to start the AR System from SunNet Manager and how to register the AR System with the SunNet Manager event stream.

AR System Documents

The ***Action Request System Installation Guide*** is a guide to installing the AR System software for all operating environments.

The ***Action Request System User's Guide*** is a how-to description of the operations performed by all users of the AR System. There are separate User's Guides for environments supporting the Motif, Macintosh, and Windows graphical user interfaces (GUIs) and for ASCII terminals.

The ***Action Request System Administrator's Guide*** describes how the AR System Administrator can use the Admin Tool to set up the AR System and define its local operations. This manual is also a reference of advanced AR System concepts. There are separate Administrator's Guides for the UNIX and Windows environments.

The ***Action Request System Programmer's Guide*** is a reference guide for programming with the application programming interfaces (APIs) that come with the AR System.

The ***Action Request System Troubleshooting and Error Messages Guide*** provides information to help you identify and solve problems with the AR System.

The ***Help Desk Templates Guide for Users and Administrators*** describes the Helpdesk application that runs in conjunction with the AR System to help you manage your internal help desk organization. The template takes full advantage of the rich feature set of the AR System and implements workflow and reporting mechanisms to simplify the task of working in or managing a help desk. Refer to this guide for installation information for the Network Management Integration options.

The ***Action Request System Workflow Demonstration and Guide to Sample Schemas*** leads you through an online demonstration showing the use of the AR System in a sample help desk environment and describes how you can use the sample schemas supplied with the AR System.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	**Meaning**	**Example**
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-2 Shell Prompts

Shell	**Prompt**
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Part 1 — Integration Options with HP OpenView

January 1996
Part Number: APP-210-DN9-P01

Integration Options with HP OpenView

This module discusses integration options for the AR System and HP OpenView. The topics covered include the following:

- Starting the AR System from HP OpenView.
- Registering with the HP OpenView Event Stream.

Starting the AR System from HP OpenView

You can run the AR System User Tool from the command line. In addition, you can specify a macro, including parameters, to be executed upon startup. (See the *Action Request System User's Guide* for information on how to create a macro and how to run a macro at startup.)

The first integration option with HP OpenView allows you to initiate the AR System while you are working within OpenView. If a machine is pictured on your OpenView map, you can invoke the AR System User Tool using a macro that finds current or past problems with that machine. The User Tool is started using the menu bar integration concept of OpenView to accomplish this goal.

Note – HP OpenView 3.0 is supported on Sun systems running SunOS 4.1.3 and Solaris 2.3 as well as on HP systems running HP-UX 9.0.

Menu Bar Integration

Within OpenView, you can integrate with the OpenView window, such as the one shown in Figure 1, by adding menus to the menu bar and/or menu items to new or existing menus. The items added can be associated with all symbol types or with any specific set of symbols you choose. The items added to a menu allow execution of an arbitrary UNIX command line (in this case, the User Tool running a macro). When the user chooses the menu item you have added, the process associated with that menu item executes.

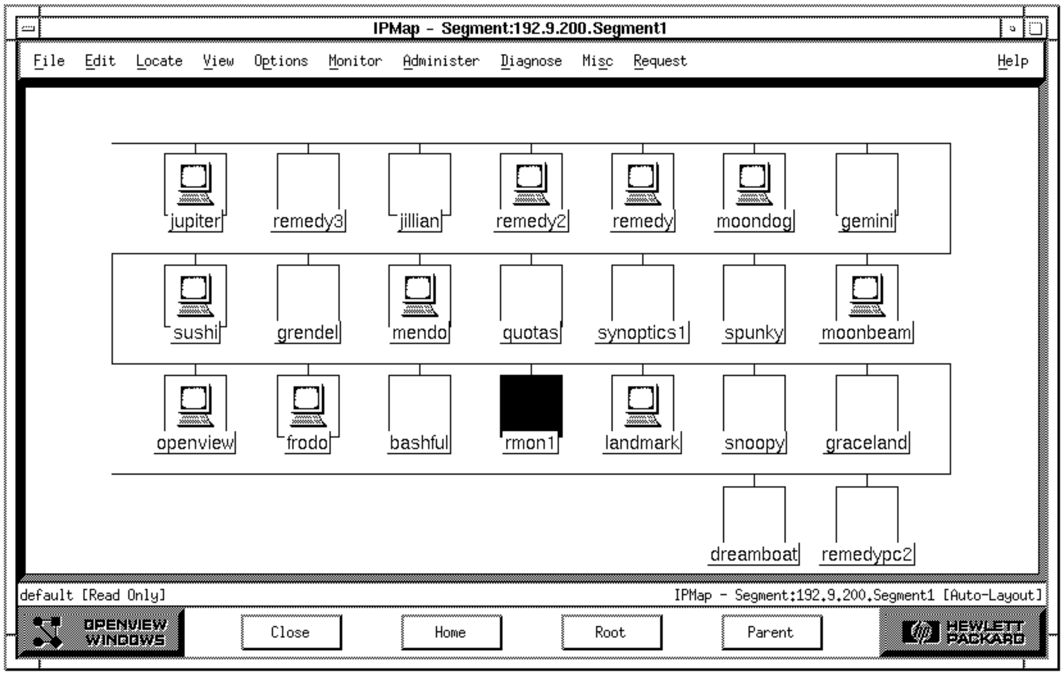


Figure 1 Open View Window

An Example Integration with the OpenView Windows Menu System

This section provides an example of how you can use the OpenView Windows menu integration feature. It does not include a complete discussion of the various commands and options you can use. For detailed information about the options available when integrating with OpenView Windows, refer to the HP OpenView documentation. We suggest you specifically read about the `OVwRegIntro(5)` option in the *HP OpenView Administrator's Reference Manual*.

By using the OpenView Windows menu integration feature in combination with the AR System command line options, you can define an HP OpenView command option that starts the User Tool with any command you desire. In addition, you can include a parameter that lists the currently selected node in the command line.

You do this by defining the command line with the parameter `{$OVwSelections}` (for all machines selected) or `{$OVwSelection1}` (for the first selection within a set of selected machines) where you want the list specified. OpenView windows substitutes the list into the command line before executing it. This is especially useful for passing values to parameters in the User Tool command line.

For example, if you have recorded a macro named OpenTickets to retrieve all open tickets for a given machine and have recorded that query with a parameter named "machine," you could retrieve all the open tickets on machine *frodo* with the following UNIX command line:

```
% aruser -i OpenTickets -p machine=frodo &
```

If you want to define a OpenView Windows menu named `AR System` that contains a menu item named `Request` that contains two pull-right sub-menu items named `Show Open Tickets` and `Submit`, you can copy the `sample.openview` file (located in the `<ar_install_dir>/bin` directory) to the OpenView registration directory. (The default location is `/usr/OV/registration/C`.) The next time OpenView Windows is started, it will contain the new menu.

Code Example 1 Open View Windows Menu

```
// OpenView Windows Application integration for AR System.

Application “AR System Integration”
{
    Version “2.0”;

    Description {
    “Enables you to specify AR System macro operations”,
    “that will be performed for the selected machine.”
    }

    Copyright {
```

Code Example 1 Open View Windows Menu *(Continued)*

```
// OpenView Windows Application integration for AR System.
    “(c) Copyright 1993-1995 Remedy Corporation. All Rights
reserved.”
    }
    MenuBar <50> “Request” _R
    {
    “AR System” f.menu “AR System”;
    }

    Menu “AR System”
    {
    <100> “Show Open Tickets...” _O f.action “Show Open”;
    <10> “Submit...” _S ! “/usr/ar/bin/aruser -s -I”;
    }

    Action “Show Open”
    {
    MinSelected 1;
    MaxSelected 1;
    Command “/usr/ar/bin/aruser -q -iOpenTickets -
pmachine=$OVwSelections”;
    }
}
```

You can change the menu definitions at any time by simply updating the definition file and restarting the OpenView Windows program.

Registering with the HP OpenView Event Stream

The second integration option between the AR System and OpenView is at the event stream. The AR System lets you track entries (ARs) as they move through various states. ARs submitted for network events that are reporting failures and exceptions on the network are good candidates for entries to be tracked and resolved.

The AR System supplies a full set of APIs for interaction with the AR System server. OpenView provides a set of APIs to allow processes to register with the event stream being processed by OpenView and to receive copies of all events received. The AR System includes a daemon that uses both of these APIs (`arovd`), as shown in Figure 2. When an event is received, the daemon decodes the event, checks against a list of events you have specified, and, if it finds a match, creates a new entry in the AR System according to the translation specifications you have supplied.

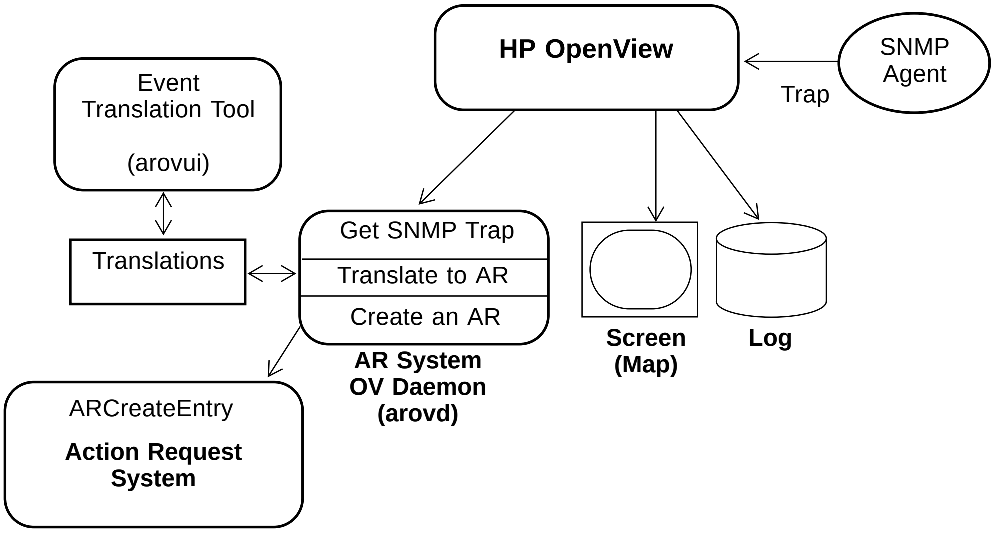


Figure 2 AR System / Open View Daemon

The AR System/OpenView Daemon (`arovd`)

The daemon process that provides the interface between the AR System and OpenView is named `arovd`. It is located at `<ar_install_dir>/bin/arovd`. In order for any events to be translated into entries within the AR system, the `arovd` process must be running. You must start this process *after* you start the OpenView daemon process, since it must register with OpenView to receive events. If you will be running in an environment where this daemon should always or often be run, it is a good idea to put together a small script that starts OpenView and then the daemon. You can then start the correct environment simply by executing the script.

An example script follows:

```
# ! /bin/csh
ovstart&
<ar_install_dir>/bin/arovd &
```

When run, arovd registers with the OpenView event stream and requests that copies of all events received be forwarded to it. It then waits on the callback procedure specified in the registration for any events forwarded. When it receives an event it decodes the event and checks to see if the event is one in which you have registered interest.

The configuration file that contains the translations you have specified is then opened for processing (see the discussion of the arovui process that follows). Since the configuration file is opened when an event arrives, any changes you made to the translations of matching events are current. The configuration file is read sequentially to find any definition that matches the received event.

Note – Because the configuration file is processed sequentially, the first match is processed regardless of whether there are later (and possibly more complete) matches in the file. You choose the order in which the events are registered in the configuration file.

If no match is found in the configuration file, the event is ignored and no further action is taken. However, if a match is found, the event is translated to the AR System using the translations you specified when you registered interest in the event (again, see the discussion of the arovui process).

The arovd daemon allows you to configure which events in your network are of interest (for example, an SNMP Cold-start trap from all routers in the network) and how the events are to be translated into the AR System (for example, into the AR-HD Calls schema). The submitter of the new entry is HP Openview so you have a record of (and can filter on) where the entry came from.

When the AR System entry is made, all the filters defined for the schema are processed just as if you had manually entered the new AR into the system. Active links, however, are ignored and no active link actions occur.

It is important to note that all entries created by the arovd daemon will be submitted by a user named HP Openview. This is important in the access control scheme of the AR System. In short, either all fields that will be loaded by the daemon must be Open at submit time or you must register a user named HP Openview with appropriate permissions to allow write access to the fields it will be loading.

If you specify a password for the HP Openview user you create, you must supply that password to the arovd process using the -p command line option. For example, to start arovd in an environment where there is an HP Openview user with a password of OVPass, you would issue the following command:

```
% <ar_install_dir>/bin/arovd -p OVPass &
```

*The AR System/OpenView Registration Tool (*arovui*)*

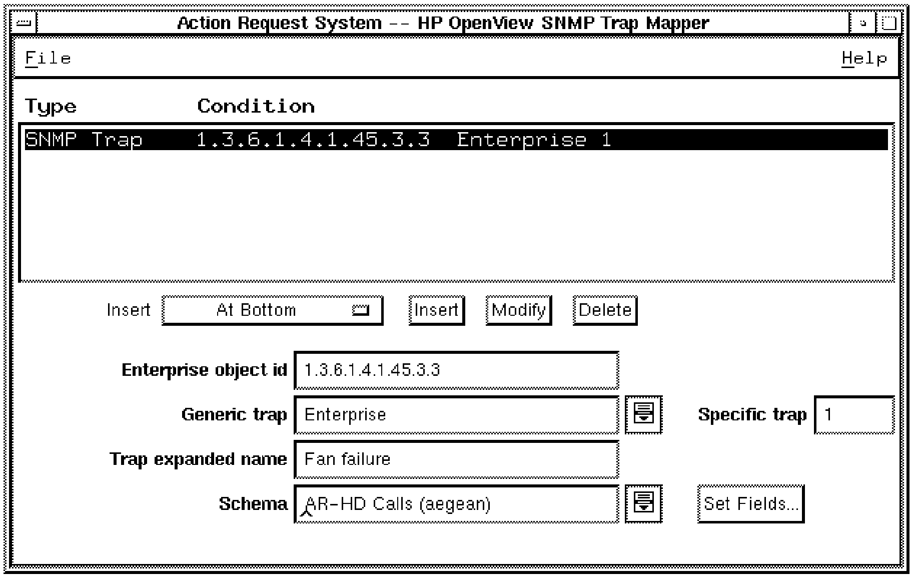
The AR System / OpenView Registration Tool provides an interface to allow you to register translations for SNMP traps you want mapped into entries in the AR System. You need to know which SNMP traps you are interested in before using this tool. You can add new or delete existing entries at any time by running this tool. Any changes take effect immediately when saved.

Logged in as root (or the user you installed the AR System software as, if you performed a non-root installation), you run the tool using the following command:

```
# <ar_install_dir>/bin/arovui &
```

Note – If you are not logged in as root, you can still run the arovui tool, but you cannot save your work unless you have write access to the <ar_install_dir>/db directory. Typically, you must be root to write to the <ar_install_dir>/db directory.

The Action Request System — HP OpenView SNMP Trap Mapper window appears, as shown in Figure 3.



Type	Condition
SNMP Trap	1.3.6.1.4.1.45.3.3 Enterprise 1

Enterprise object id: 1.3.6.1.4.1.45.3.3

Generic trap: Enterprise

Specific trap: 1

Trap expanded name: Fan failure

Schema: AR-HD Calls (aegean)

Figure 3 AR System - HP Open View SNMP Trapper Window

At startup, the process registers as user `HP Openview`. This is done so that any translations that are defined are legal when the `arovd` daemon specifies them.

Like the `arovd` daemon, if there is a password for the `HP Openview` user, you must specify the password using the `-p` command line option at startup. If you do not supply a required password, you will not be able to create new translations.

For example, to start `arovui` with user `HP Openview` having a password of `OVPass`, enter the following command:

```
#<ar_install_dir>/bin/arovui -p OVPass &
```

File Menu

The Action Request System — HP OpenView SNMP Trap Mapper window File menu provides the following options:

- **Save -** Saves the changes you have defined to the file `<ar_install_dir>/db/ov.ar`. You can also enter `Ctrl+S`.
- **Exit -** Exits the tool. You can also enter `Alt+F4`.

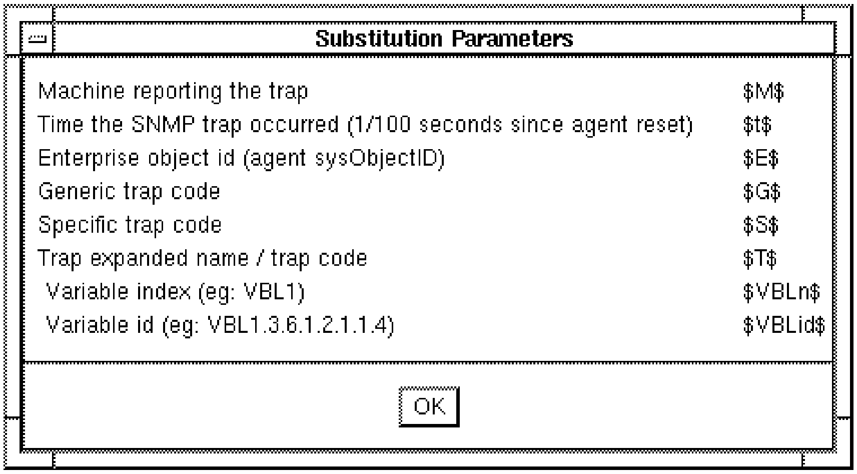
Help Menu

The Help menu provides the following options:

- **On Substitution Parameters -** Brings up the Substitution Parameters window (shown in Figure 4) displaying the parameters available for translating information. You can also enter `Ctrl+P`.
- **On Version -** Supplies information about the tool, such as the version number and on which client machine the tool is running. You can also enter `Ctrl+V`.

Substitution Parameters

You can bring up a display of the parameters you can specify to translate information into the various fields of a new entry by selecting Substitution Parameters from the Help menu. The Substitution Parameters window appears, as shown in Figure 4.



Machine reporting the trap	M
Time the SNMP trap occurred (1/100 seconds since agent reset)	t
Enterprise object id (agent sysObjectID)	E
Generic trap code	G
Specific trap code	S
Trap expanded name / trap code	T
Variable index (eg: VBL1)	$VBLn$
Variable id (eg: VBL1.3.6.1.2.1.1.4)	$VBLid$

Figure 4 Substitution Parameters Window

You can use any of the available parameters in any mapping. For `VarBindList` variables, if the specified variable does not exist, a null value is used for the translation.

Table 1 describes all the parameters you can specify to translate information into the various fields of a new entry:

Note – The variable bindings list (`VarBindList`) is a list of variables that are defined to hold specific information about SNMP traps. The `VarBindList` variables in Table 1 begin with the characters `$VBL`.

Table 1 Substitution Parameters

Parameter	Value
M	Network address of the agent (machine) reporting the trap.
t	The time that the SNMP trap occurred. This is not absolute time. It is 1/100 seconds since the agent was reset.
E	The agent's System Object ID.
G	The generic trap code.
S	The enterprise specific trap code.
T	The expanded name of the trap. If the expanded name is not defined, the default is the trap code.
$VBL<*n*>$	Value of the `VarBindList` variable at location <*n*> (where *n* is the index to the `VarBindList`). Zero (0) is the index to the first variable.
$VBL<*id*>$	Value of the `VarBindList` variable with an object identifier equal to <*id*>.

Modifying Registrations

The listbox at the top of the Action Request System — HP OpenView SNMP Trap Mapper window (shown in Figure 3 on page 8) contains an ordered list of all of the events already registered. It displays the type of the event (currently only SNMP trap) along with any specified conditions.

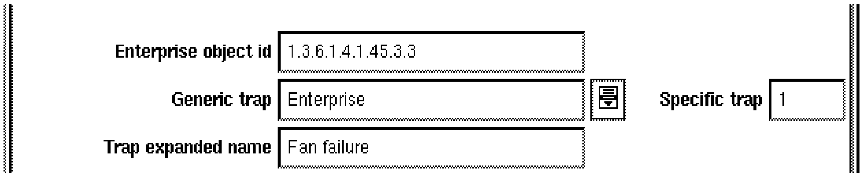
The order of items in the listbox is important as they are processed serially. The first match in the file is chosen regardless of whether or not it is the best match. This is especially important to remember if wild cards are used in the conditions.

You can specify the following options:

- **Insert -** Inserts a new item in the listbox built from the information on the lower half of the screen. You can choose from the selection menu next to the Insert button to specify whether the item should appear at the top of the list, at the bottom, or before or after the currently selected item.
- **Modify -** Modifies the selected item to the new information in the lower half of the window.
- **Delete -** Deletes the currently selected item from the list.

Matching an SNMP Trap

Matching an SNMP trap involves checking the enterprise ID of the object that generated the trap and the type of trap generated. You specify the match parameters by entering values in the middle of the Action Request System — HP OpenView SNMP Trap Mapper window, as shown in Figure 5.



Enterprise object id: 1.3.6.1.4.1.45.3.3
Generic trap: Enterprise
Specific trap: 1
Trap expanded name: Fan failure

Figure 5 Matching an SNMP Trap

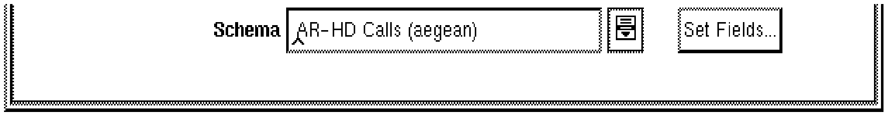
- **Enterprise object id -** The enterprise registration ID of the agent you are interested in. If you want to match a class of traps, you can use an asterisk (*) at the end of the ID to specify any trap matching the registration ID to this point.

  Note that the order of registration in the listbox is important when using an asterisk (*) to ensure any specific matches (such as 1.3.6.1.4.1.400.5.5) occur before any wild card matches (such as 1.3.6.1.4.1.400.*) or the specific match will always be superceded by the wild card match.

- **Generic trap -** The string identifying the trap type. This string is one of the values included on the menu associated with this field. Select an item from the menu to enter it in the field. If you want to match all types of traps, select All from the menu (or simply leave the field blank). An asterisk (*) is the wild card character.
- **Specific trap -** If the value you enter in the Generic trap field is Enterprise, you must enter the enterprise code number for the trap in this field. An asterisk (*) is the wild card character. (If you enter any value other than Enterprise in the Generic trap field, this field is grayed.)
- **Trap expanded name -** If you have not specified a mapping, or you want to remap the trap type string to something more readable or meaningful, enter the string you want used for any trap type (`$T$`) substitutions in this field. This field is especially useful for mapping enterprise-specific traps to a friendlier name.

Translating an Event into an AR System Entry

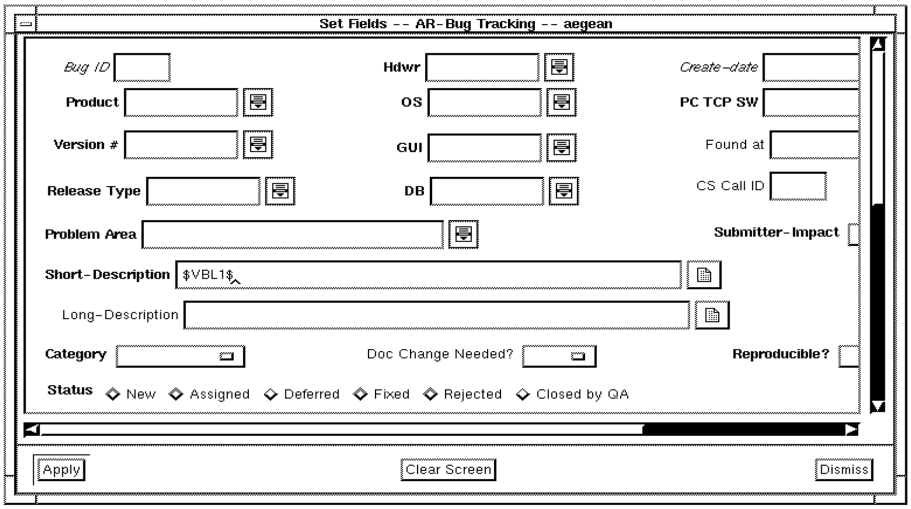
You specify the translation to the AR System using the final fields on the Action Request System — HP OpenView SNMP Trap Mapper window, as shown in Figure 6.



Schema AR-HD Calls (aegean)

Figure 6 Translating an Event to an AR System Entry

- **Schema -** The name of the schema to translate into. The menu contains a list of all the accessible schemas. You simply select the menu button to the right of the field then select from the list to enter a schema name in the field.
- **Set Fields -** The set fields button brings up a window containing all the fields for the specified schema that the `HP Openview` user can set at submit time. An example of the Set Fields window is shown in Figure 7.



Short-Description: $VBL1$

Figure 7 Set Fields Window for OpenView Event Translation

All the fields set automatically by the AR System and all fields loaded automatically by `arovd` are read only. You can specify any text (including parameters) for the other fields. If no value is specified for any of the fields, no value is supplied when the entry is created.

By default, the Short-Description field (field ID=8) is set to the following:

- **SNMP trap**. SNMP Trap — `$M$: $E$: $T$`

This provides a consistent style and format for traps/events mapped from the OpenView event stream.

You can change the value for this field to any value desired. Your value will override the default.

Be aware of the following issues:

- You will not be able to open any Text Edit or Diary Entry windows from the Set Fields window.
- You cannot use AR System keywords in the mapping.

- Query-style menus that reference values on the current screen will produce an error message regarding inconsistent data types, and will not work in the Set Fields window.

Part 2 — Integration Options with IBM NetView/6000

January 1996
Part Number: APP-210-DN9-N01

Integration Options with IBM NetView/6000

This module discusses integration options for the AR System and IBM NetView / 6000. The topics covered include the following:

- Starting the AR System from IBM NetView / 6000.
- Registering with the IBM NetView / 6000 Event Stream.

Starting the AR System from IBM NetView/6000

You can run the AR System User Tool from the command line. In addition, you can specify a macro, including parameters, to be executed upon startup. (See the *Action Request System User's Guide* for information on how to create a macro and how to run a macro at startup.)

The first integration option with IBM NetView allows you to initiate the AR System while you are working within NetView. If a machine is pictured on your NetView map, you can invoke the AR System User Tool using a macro that finds current or past problems with that machine. The User Tool is started using the menu bar integration concept of NetView to accomplish this goal.

Menu Bar Integration

Within NetView, you can integrate with the NetView window, such as the one shown in Figure 8, by adding menus to the menu bar and / or menu items to new or existing menus. The items added can be associated with all symbol types or with any specific set of symbols you choose. The items added to a menu allow execution of an arbitrary UNIX command line (in this case, the User Tool running a macro). When the user chooses the menu item you have added, the process associated with that menu item executes.

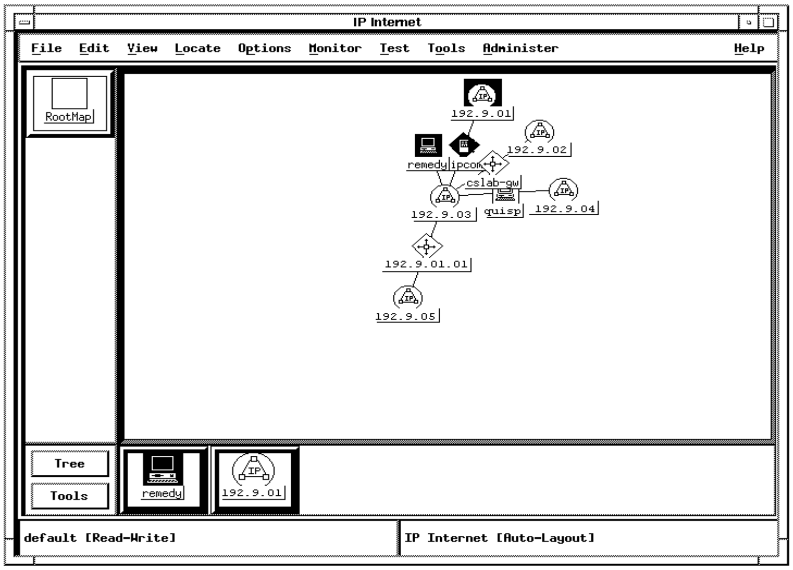


Figure 8 NetView Window

An Example Integration with the NetView Windows Menu System

This section provides an example of how you can use the NetView Windows menu integration feature. It does not include a complete discussion of the various commands and options you can use. For detailed information about the options available when integrating with NetView Windows, refer to the IBM NetView documentation. We suggest you specifically read about the `NVwRegIntro(5)` option in the *IBM NetView Administrator's Reference Manual*.

By using the NetView Windows menu integration feature in combination with the AR System command line options, you can define an IBM NetView command option that starts the User Tool with any command you desire. In addition, you can include a parameter that lists the currently selected node in the command line.

You do this by defining the command line with the parameter `{$NVwSelections}` (for all machines selected) or `{$NVwSelection1}` (for the first selection within a set of selected machines) where you want the list specified. NetView Windows substitutes the list into the command line before executing it. This is especially useful for passing values to parameters in the User Tool command line.

For example, if you have recorded a macro named OpenTickets to retrieve all open tickets for a given machine and have recorded that query with a parameter named “machine,” you could retrieve all the open tickets on machine frodo with the following UNIX command line:

```
% aruser -i OpenTickets -p machine=frodo &
```

If you want to define a NetView Windows menu named AR System that contains a menu item named Request that contains two pull-right sub-menu items named Show Open Tickets and Submit, you can modify the sample.openview file (located in the <ar_install_dir>/bin directory). Then you can copy the modified file to the NetView registration directory. (The default location is /usr/NV/registration/C.) The next time NetView Windows is started, it will contain the new menu.

Code Example 2 NetView Windows Menu

```
// NetView Windows Application integration for AR System.

Application “AR System Integration”
{
    Version “2.0”;

    Description {
    “Enables you to specify AR System macro operations”,
    “that will be performed for the selected machine.”
    }

    Copyright {
    “(c) Copyright 1993-1995 Remedy Corporation. All Rights
reserved.”
    }
    MenuBar <50> “Request” _R
    {
    “AR System” f.menu “AR System”;
    }

    Menu “AR System”
    {
```

Code Example 2 NetView Windows Menu *(Continued)*

```
// NetView Windows Application integration for AR System.
    <100> “Show Open Tickets...” _O f.action “Show Open”;
    <10> “Submit...” _S ! “/usr/ar/bin/aruser -s -I”;
    }

    Action “Show Open”
    {
    MinSelected 1;
    MaxSelected 1;
    Command “/usr/ar/bin/aruser -q -iOpenTickets -
pmachine=$NVwSelections”;
    }
}
```

You can change the menu definitions at any time by simply updating the definition file and restarting the NetView Windows program.

Registering with the IBM NetView/6000 Event Stream

The second integration option between the AR System and NetView is at the event stream. The AR System lets you track entries (ARs) as they move through various states. ARs submitted for network events that are reporting failures and exceptions on the network are good candidates for entries to be tracked and resolved.

The AR System supplies a full set of APIs for interaction with the AR System server. NetView provides a set of APIs to allow processes to register with the event stream being processed by NetView and to receive copies of all events received. The AR System includes a daemon that uses both of these APIs (`arnvd`), as shown in Figure 9. When an event is received, the daemon decodes the event, checks against a list of events you have specified, and, if it finds a match, creates a new entry in the AR System according to the translation specifications you have supplied.

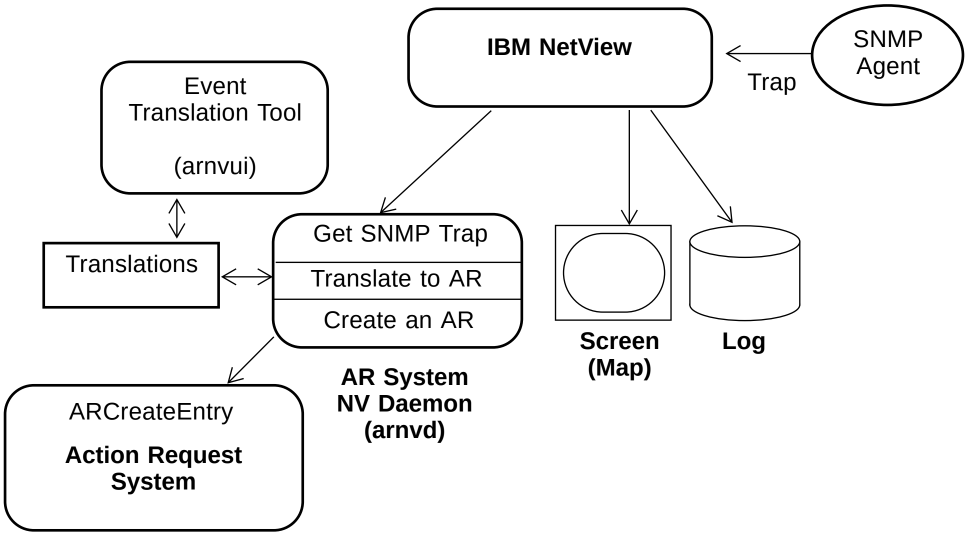


Figure 9 AR System / NetView Daemon

The AR System/NetView Daemon (arnvd)

The daemon process that provides the interface between the AR System and NetView is named `arnvd`. It is located at `<ar_install_dir>/bin/arnvd`. In order for any events to be translated into entries within the AR System, the `arnvd` process must be running. You must start this process *after* you start the NetView daemon process, since it must register with NetView to receive events. If you will be running in an environment where this daemon should always or often be run, it is a good idea to put together a small script that starts NetView and then the daemon. You can then start the correct environment simply by executing the script.

An example script follows:

```
# ! /bin/csh
nvstart&
<ar_install_dir>/bin/arnvd &
```

When run, arnvd registers with the NetView event stream and requests that copies of all events received be forwarded to it. It then waits on the callback procedure specified in the registration for any events forwarded. When it receives an event it decodes the event and checks to see if the event is one in which you have registered interest.

The configuration file that holds the translations you have specified is then opened for processing (see the discussion of the arnvui process that follows). Since the configuration file is opened when an event arrives, any changes you made to the translations of matching events are current. The configuration file is read sequentially to find any definition that matches the received event.

Note – Because the configuration file is processed sequentially, the first match is processed regardless of whether there are later (and possibly more complete) matches in the file. You choose the order in which the events are registered in the configuration file.

If no match is found in the configuration file, the event is ignored and no further action is taken. However, if a match is found, the event is translated to the AR System using the translations you specified when you registered interest in the event (again, see the discussion of the arnvui process).

The arnvd daemon allows you to configure which events in your network are of interest (for example, an SNMP Cold-start trap from all routers in the network) and how the events are to be translated into the AR System (for example, into the AR-HD Calls schema). The submitter of the new entry is IBM Netview so you have a record of (and can filter on) where the entry came from.

When the AR System entry is made, all the filters defined for the schema are processed just as if you had manually entered the new AR into the system. Active links, however, are ignored and no active link actions occur.

It is important to note that all entries created by the arnvd daemon will be submitted by a user named IBM Netview. This is important in the access control scheme of the AR System. In short, either all fields that will be loaded by the daemon must be Open at submit time or you must register a user named IBM Netview with appropriate permissions to allow write access to the fields it will be loading.

If you specify a password for the IBM Netview user you create, you must supply that password to the arnvd process using the -p command line option. For example, to start arnvd in an environment where there is a IBM Netview user with a password of NVPass, you would issue the following command:

```
% <ar_install_dir>/bin/arnvd -p NVPass &
```

The AR System/NetView Registration Tool (arnvui)

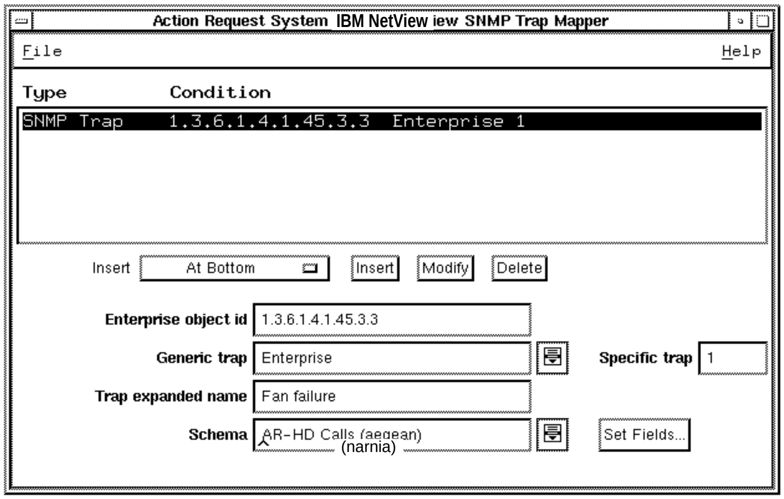
The AR System / NetView Registration Tool provides an interface to allow you to register translations for SNMP traps you want mapped into entries in the AR System. You need to know which SNMP traps you are interested in before using this tool. You can add new or delete existing entries at any time by running this tool. Any changes take effect immediately when saved.

Logged in as root (or the user you installed the AR System software as, if you performed a non-root installation), you run the tool using the following command:

```
# <ar_install_dir>/bin/arnvui &
```

Note – If you are not logged in as root, you can still run the arnvui tool, but you cannot save your work unless you have write access to the <ar_install_dir>/db directory. Typically, you must be root to write to the <ar_install_dir>/db directory.

The Action Request System — IBM NetView SNMP Trap Mapper window appears, as shown in Figure 10.



Type	Condition
SNMP Trap	1.3.6.1.4.1.45.3.3 Enterprise 1

Enterprise object id: 1.3.6.1.4.1.45.3.3

Generic trap: Enterprise

Specific trap: 1

Trap expanded name: Fan failure

Schema: AR-HD Calls (aegean) (narnia)

Figure 10 AR System - IBM NetView SNMP Trap Mapper Window

At startup, the process registers as user `IBM Netview`. This is done so that any translations that are defined are legal when the `arnvd` daemon specifies them.

Like the `arnvd` daemon, if there is a password for the `IBM Netview` user, you must specify the password using the `-p` command line option at startup. If you do not supply a required password, you will not be able to create new translations.

For example, to start `arnvui` with user `IBM Netview` having a password of `NVPass`, enter the following command:

```
#<ar_install_dir>/bin/arnvui -p NVPass &
```

File Menu

The Action Request System — IMB NetView SNMP Trap Mapper window File menu provides the following options:

- **Save -** Saves the changes you have defined to the file `<ar_install_dir>/db/ov.ar`. You can also enter `Ctrl+S`.

- **Exit -** Exits the tool. You can also enter `Alt+F4`.

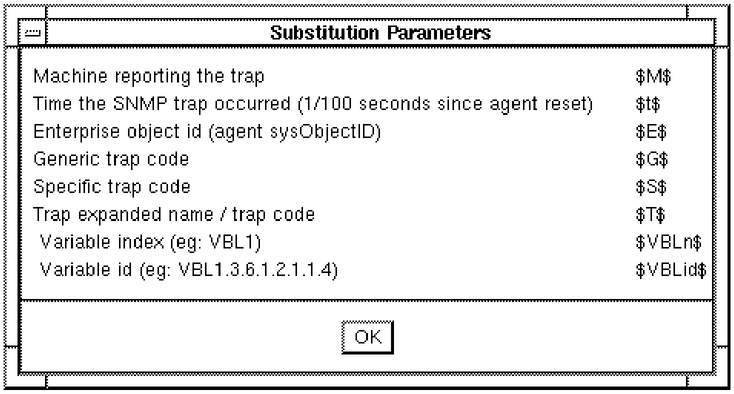
Help Menu

The Help menu provides the following options:

- **On Substitution Parameters -** Brings up the Substitution Parameters window (shown in Figure 11) displaying the parameters available for translating information. You can also enter `Ctrl+P`.
- **On Version -** Supplies information about the tool, such as the version number and on which client machine the tool is running. You can also enter `Ctrl+V`.

Substitution Parameters

You can bring up a display of the parameters you can specify to translate information into the various fields of a new entry by selecting Substitution Parameters from the Help menu. The Substitution Parameters window appears, as shown in Figure 11.



Machine reporting the trap	M
Time the SNMP trap occurred (1/100 seconds since agent reset)	t
Enterprise object id (agent sysObjectID)	E
Generic trap code	G
Specific trap code	S
Trap expanded name / trap code	T
Variable index (eg: VBL1)	$VBLn$
Variable id (eg: VBL1.3.6.1.2.1.1.4)	$VBLid$

Figure 11 Substitution Parameters Window

You can use any of the available parameters in any mapping. For `VarBindList` variables, if the specified variable does not exist, a null value is used for the translation.

Table 2 describes all the parameters you can specify to translate information into the various fields of a new entry:

Note – The variable bindings list (`VarBindList`) is a list of variables that are defined to hold specific information about SNMP traps. The `VarBindList` variables in Table 2 begin with the characters `$VBL`.

Table 2 Substitution Parameters

Parameter	Value
M	Network address of the agent (machine) reporting the trap.
t	The time that the SNMP trap occurred. This is not absolute time. It is 1/100 seconds since the agent was reset.
E	The agent's System Object ID.
G	The generic trap code.
S	The enterprise specific trap code.
T	The expanded name of the trap. If the expanded name is not defined, the default is the trap code.
$VBL<*n*>$	Value of the `VarBindList` variable at location <*n*> (where *n* is the index to the `VarBindList`). Zero (0) is the index to the first variable.
$VBL<*id*>$	Value of the `VarBindList` variable with an object identifier equal to <*id*>.

Modifying Registrations

The listbox at the top of the Action Request System — IBM NetView SNMP Trap Mapper window (Figure 10 on page 24) contains an ordered list of all of the events already registered. It displays the type of the event (currently only SNMP trap) along with any specified conditions.

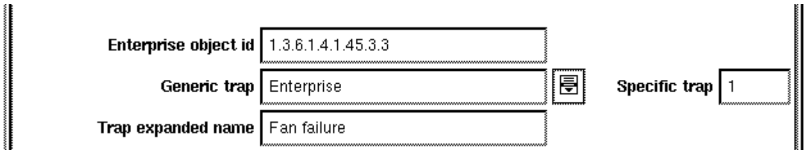
The order of items in the listbox is important as they are processed serially. The first match in the file is chosen regardless of whether or not it is the best match. This is especially important to remember if wild cards are used in the conditions.

You can specify the following options:

- **Insert -** Inserts a new item in the listbox built from the information on the lower half of the screen. You can choose from the selection menu next to the Insert button to specify whether the item should appear at the top of the list, at the bottom, or before or after the currently selected item.
- **Modify -** Modifies the selected item to the new information in the lower half of the window.
- **Delete -** Deletes the currently selected item from the list.

Matching an SNMP Trap

Matching an SNMP trap involves checking the enterprise ID of the object that generated the trap and the type of trap generated. You specify the match parameters by entering values in the middle of the Action Request System — IBM NetView SNMP Trap Mapper window, as shown in Figure 12.



Enterprise object id: 1.3.6.1.4.1.45.3.3
Generic trap: Enterprise
Specific trap: 1
Trap expanded name: Fan failure

Figure 12 Matching an SNMP Trap

- **Enterprise object id -** The enterprise registration ID of the agent you are interested in. If you want to match a class of traps, you can use an asterisk (*) at the end of the ID to specify any trap matching the registration ID to this point.

  Note that the order of registration in the listbox is important when using an asterisk (*) to ensure any specific matches (such as 1.3.6.1.4.1.400.5.5) occur before any wild card matches (such as 1.3.6.1.4.1.400.*) or the specific match will always be superceded by the wild card match.
- **Generic trap -** The string identifying the trap type. This string is one of the values included on the menu associated with this field. Select an item from the menu to enter it in the field. If you want to match all types of traps, select All from the menu (or simply leave the field blank). An asterisk (*) is the wild card character.

- **Specific trap -** If the value you enter in the Generic trap field is Enterprise, you must enter the enterprise code number for the trap in this field. An asterisk (*) is the wild card character. (If you enter any value other than Enterprise in the Generic trap field, this field is grayed.)
- **Trap expanded name -** If you have not specified a mapping, or you want to remap the trap type string to something more readable or meaningful, enter the string you want used for any trap type (`$T$`) substitutions in this field. This field is especially useful for mapping enterprise-specific traps to a friendlier name.

Translating an Event into an AR System Entry

You specify the translation to the AR System using the final fields on the Action Request System — IBM NetView SNMP Trap Mapper window, as shown in Figure 13.

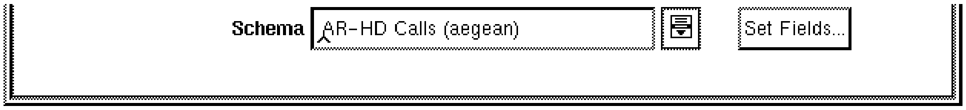
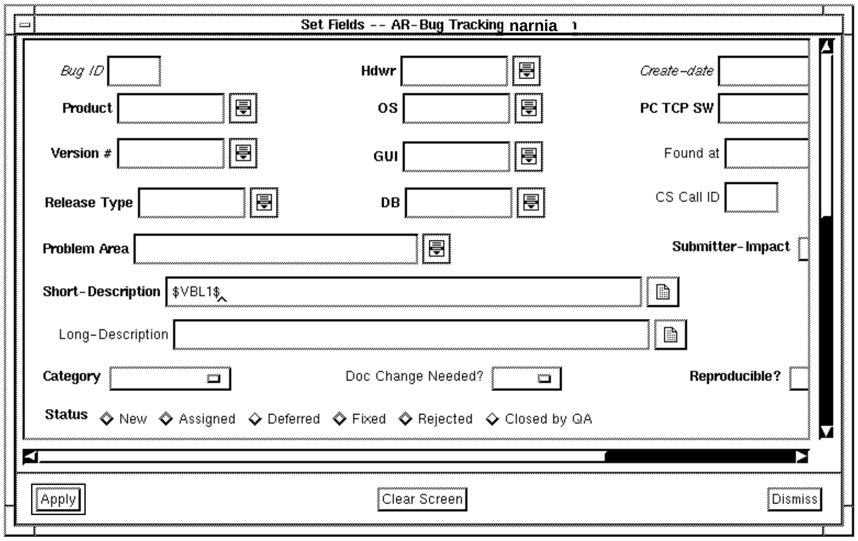


Figure 13 Translating an Event to an AR System Entry

- **Schema -** The name of the schema to translate into. The menu contains a list of all the accessible schemas. You simply select the menu button to the right of the field then select from the list to enter a schema name in the field.
- **Set Fields -** The set fields button brings up a window containing all the fields for the specified schema that the `IBM Netview` user can set at submit time. An example of the Set Fields window is shown in Figure 14.



Short-Description: $VBL1$

Figure 14 Set Fields Window for NetView Event Translation

All the fields set automatically by the AR System and all fields loaded automatically by `arnvd` are read only. You can specify any text (including parameters) for the other fields. If no value is specified for any of the fields, no value is supplied when the entry is created.

By default, the Short-Description field (field ID=8) is set to the following:

- **SNMP trap**. SNMP Trap — `$M$: $E$: $T$`

This provides a consistent style and format for traps/events mapped from the NetView event stream.

You can change the value for this field to any value desired. Your value will override the default.

Be aware of the following issues:

- You will not be able to open any Text Edit or Diary Entry windows from the Set Fields window.
- You cannot use AR System keywords in the mapping.

- Query-style menus that reference values on the current screen will produce an error message regarding inconsistent data types, and will not work in the Set Fields window.

Part 3 — Integration Options with SunNet Manager

January 1996
Part Number: APP-210-DN9-S01

Integration Options with SunNet Manager

This module discusses integration options for the AR System and SunNet Manager. The topics covered include the following:

- Starting the AR System from SunNet Manager.
- Registering with the SunNet Manager Event Stream.

Starting the AR System from SunNet Manager

You can run the AR System User Tool from the command line. In addition, you can specify a macro, including parameters, to be executed upon startup. (See the *Action Request System User's Guide* for information on how to create a macro and how to run a macro at startup.)

The first integration option with SunNet Manager (SNM) allows you to initiate the AR System while you are working within SNM. If a machine is pictured on your SNM map, you can invoke the AR System User Tool using a macro that finds current or past problems with that machine. The User Tool is started using the menu bar integration concept of SNM to accomplish this goal.

Using User Commands

Within SNM, there is a concept of User Commands connected with the Tools menu. In the SNM map, when you choose the Menu button on any glyph, a menu is displayed. In addition to other items, there is an item labeled Tools, as shown in Figure 15. These commands are simply UNIX command lines with menu tags that are attached to one or more classes of glyphs. The menu under Tools displays all of the currently configured User Commands in the system. Refer to the discussion of User Commands in the SunNet Manager documentation for a complete discussion.

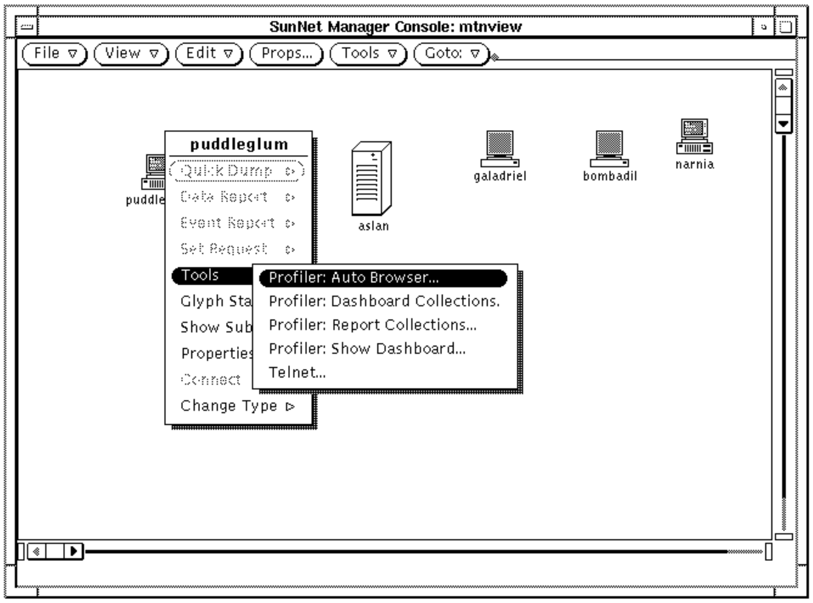


Figure 15 SunNet Manager User Commands Display

Using this feature in combination with the AR System command line options, you can define a command option to start the User Tool with any command you desire. In addition, the User Command system of SNM allows you to include any of the properties of the selected glyph into the command line. You do this by defining the command line with the name of the property preceded by a percent sign (%). SNM will substitute the value of the property into the command line before execution. This ability is especially useful for loading values for parameters in the User Tool command line.

For example, if you have recorded a macro named OpenTickets to retrieve all open tickets for a given machine and recorded that query with a parameter named machine, you could retrieve all the open tickets on machine `frodo` by entering the following command from the UNIX command line:

```
% aruser -i OpenTickets -p machine=frodo &
```

You could also define an SNM User Command. For a User Command called `Show Open Tickets`, attached to all Sun4 type machines, the entry in one of the SNM schema files would look like the following:

```
instance elementCommand (
(component.sun4 "Show Open Tickets" "aruser -i OpenTickets -p
machine=%Name")
)
```

The `instance elementCommand` clause is used to define User Command items. In the definition line itself, `component.sun4` identifies the class of glyphs with which to associate the command, with the second field being the title and the third the command to execute. The `%Name` definition at the end of the command is an SNM parameter that substitutes the name of the glyph into the command line before the command is executed. This command then appears under the Tools menu.

SNM User Command Definition Tool (addsnm)

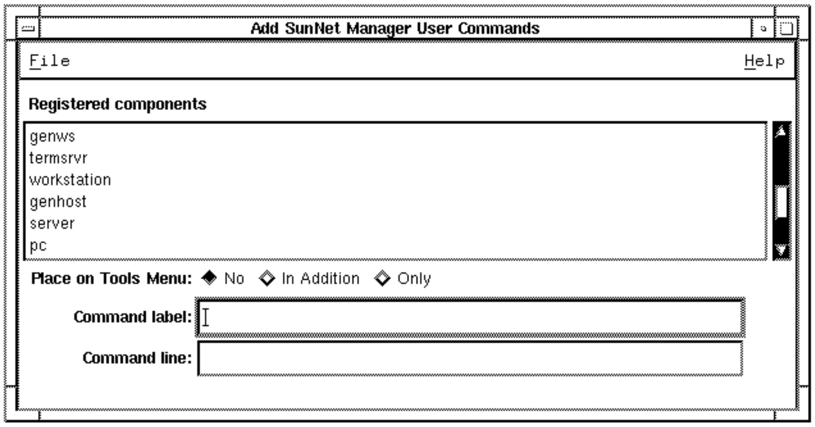
Remedy Corporation supplies the SNM User Command Definition Tool, `addsnm`, to simplify the task of adding a command in the appropriate places within SNM's configuration file.

The `addsnm` utility allows you to add one or more User Commands to one or more glyph classes in one or more SNM schema files. By default, the tool connects with the `elements.schema` file in the default location for SNM (`/usr/snm/struct/elements.schema`).

You run the tool issuing the following command:

```
% <ar_install_dir/bin/addsnm &
```

The Add SunNet Manager User Commands window appears, as shown in Figure 16.



Add SunNet Manager User Commands

Registered components

genws
termsrvr
workstation
genhost
server
pc

Place on Tools Menu: ◆ No ◇ In Addition ◇ Only

Command label:

Command line:

Figure 16 Add SunNet Manager User Commands Window

File

The Add SunNet Manager User Commands window File menu provides the following options:

- **Open -** Specify the schema file to update (default: `elements.schema`). You can change to any other file at any time. You can also enter `Ctrl+O`.
- **Save -** Save the new User Command you have defined. You can also enter `Ctrl+S`.
- **Exit -** Exit from the tool. You can also enter `Alt+F4`.

Help

The Help menu provides the following option:

- **On Version -** View information about the tool, such as the version number and on which client machine the tool is running. You can also enter `Ctrl+V`.

Defining a New User Command

Use the following fields in the Add SunNet Manager User Commands window to define the command and save the schema file.

- **Registered components -** Scrollable list of the components currently registered in the open schema file. You can select one or more of the components to identify the ones to which the user command should be added. The command is added to *all* components if none are selected.
- **Place on Tools Menu -** Select an option to specify whether to place the new command on the SNM tools menu. You can specify:
    - **No -** Do not add the command to the tools menu.
    - **In Addition -** Place the command on the tools menu as well as on the selected components.
    - **Only -** Place the command on the tools menu only, not on any components. ("Registered components" grays when you select this option.)
- **Command label -** The menu label for the command that is placed on the User Command menu.
- **Command line -** The UNIX command line that is to be executed when the label is selected. The command line cannot contain any quotes and should contain any SNM parameter definitions you want specified.

The SNM User Command Definition Tool simply updates the SNM configuration files; it does not cause SNM to recognize the changes.

To add the user command definitions to SNM:

1. Run the `addsnm` program and make whatever User Command addition(s) you desire.

```
% <ar_install_dir>/bin/addsnm &
```

2. Run SunNet Manager (or open a running copy).

```
% snm &
```

3. Choose the Load/Save option from the File menu, enter a temporary filename (for example, `/tmp/snm.map`), and choose the Save button. Be careful not to choose the Load button or you may overwrite your map!

4. Shut down SunNet Manager by choosing the Quit item from the window menu.

5. Restart SunNet Manager with the `-i` option to cause it to reread its configuration files.

```
% snm -i &
```

6. Choose the Load/Save option from the File menu, enter the name of the temporary file from step 3, and choose the Load button. Be careful not to choose the Save button or you may overwrite your saved SNM map and not be able to recover it.

7. Test that the user commands work correctly by choosing the new Tools option from one of the glyphs with which you have registered a new command.

Registering with the SunNet Manager Event Stream

The second integration option between the AR System and SNM is at the event stream. The AR System lets you track entries (ARs) as they move through various states. ARs submitted for network events that are reporting failures and exceptions on the network are good candidates for entries that need to be tracked and resolved.

The AR System supplies a full set of APIs for interaction with the AR System server. SNM provides a set of APIs to allow processes to register with the event stream being processed by SNM and to receive copies of all events received. The AR System includes a daemon that uses both of these APIs (arsnmd), as shown in Figure 17. When an event is received, the daemon decodes the event, checks against a list of events you have specified, and, if it finds a match, creates a new entry in the AR System according to the translation specifications you have supplied.

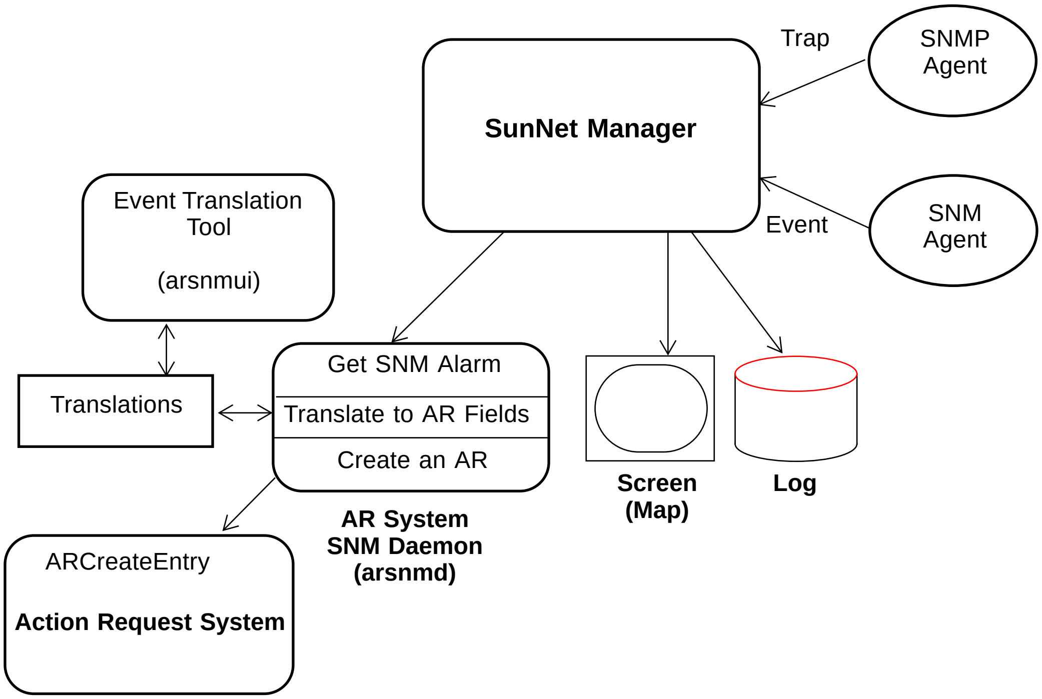


Figure 17 AR System / SNM Daemon

The AR System/SNM daemon (arsnmd)

The daemon process that provides the interface between the AR System and SNM is named `arsnmd`. It is located at `<ar_install_dir>/bin/arsnmd`. In order for any events to be translated into entries within the AR system, the `arsnmd` process must be running. In addition, you must start this process *after* you start the SNM process, since it must register with SNM to receive events. If you will be running in an environment where this daemon should always or often be run, it is a good idea to put together a small script that starts SNM and then the daemon. You can then start the correct environment simply by executing the script.

An sample script follows:

```
#! /bin/csh
snm &
<ar_install_dir>/bin/arsnmd &
```

When run, `arsnmd` registers with the SNM event stream and requests that copies of all events received be forwarded to it. It then waits on the callback procedure specified in the registration for any events forwarded. When it receives an event it decodes the event (as either an SNMP trap or a SNM event), and checks to see if the event is one in which you have registered interest.

The configuration file that holds the translations you have specified is then opened for processing (see the discussion of the `arsnmui` process that follows). Since the configuration file is opened when an event arrives, any changes you made to the translations of matching events are current. The configuration file is read sequentially to find any definition that matches the received event.

Note – Because the configuration file is processed sequentially, the first match is processed regardless of whether there are later (and possibly more complete) matches in the file. You choose the order in which the events are registered in the configuration file.

If no match is found in the configuration file, the event is ignored and no further action is taken. However, if a match is found, the event is translated to the AR System using the translations you specified when you registered interest in the event (again, see the discussion of the `arsnmui` process).

The `arsnmd` daemon allows you to configure which events in your network are of interest (for example, an SNMP Cold-start trap from all routers in the network) and how the events are to be translated into the AR System (for example, into the TroubleTicket schema). The submitter of the new entry is `SunNet Manager` so you have a record of (and can filter on) where the entry came from.

When the AR System entry is made, all the filters defined for the schema are processed just as if you had manually entered the new AR into the system. Active links, however, are ignored and no active link actions occur.

It is important to note that all entries created by the arsnmd daemon will be submitted by a user named SunNet Manager. This is important in the access control scheme of the AR System. In short, either all fields that will be loaded by the daemon must be Open at submit time or you must register a user named SunNet Manager with appropriate permissions to allow write access to the fields it will be loading.

If you specify a password for the SunNet Manager user you create, you must supply that password to the arsnmd process using the -p command line option. For example, to start arsnmd in an environment where there is a SunNet Manager user with a password of SNMPass, you would issue the following command:

```
% <ar_install_dir>/bin/arsnmd -p SNMPass &
```

The AR System/SNM Registration Tool (arsnmui)

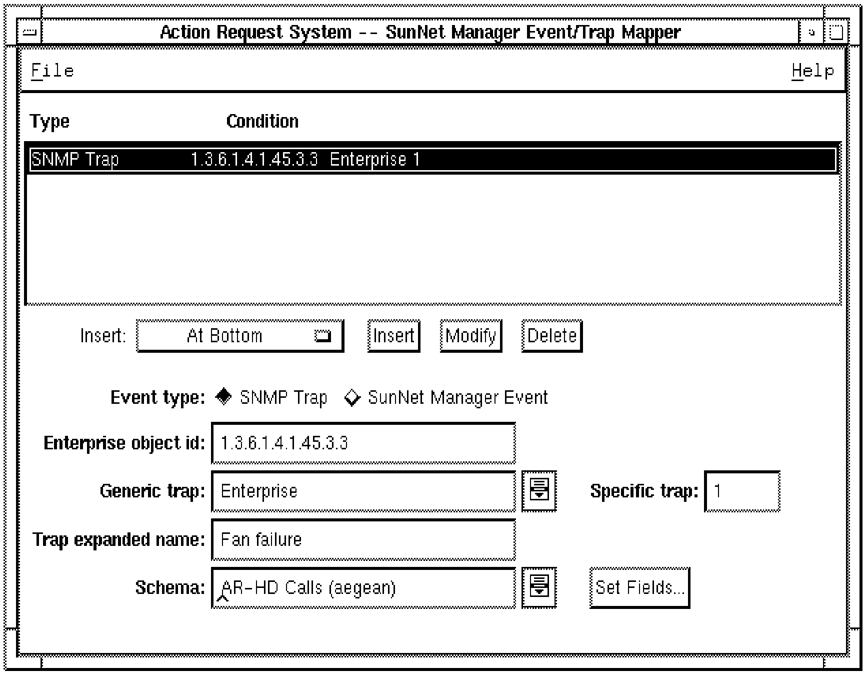
The AR System / SNM Registration Tool provides an interface that allows you to register translations for SNMP traps and SNM events that you want mapped into entries in the AR System. You will need to know which SNMP traps and SNM events you are interested in before using this tool. You can add new entries, modify existing entries, or delete existing entries at any time by running this tool. Any changes take affect immediately when saved.

Logged in as root (or the user you installed the AR System as, if you performed a non-root installation), you run the tool using the command:

```
# <ar_install_dir>/bin/arsnmui &
```

Note – If you are not logged in as root, you can still run the arsnmui tool, but you cannot save your work unless you have write access to the <ar_install_dir>/db directory. Typically, you must be root to write to the <ar_install_dir>/db directory.

The Action Request System — SunNet Manager Event / Trap Mapper window appears, as shown in Figure 18.



Type	Condition
SNMP Trap	1.3.6.1.4.1.45.3.3 Enterprise 1

Insert: At Bottom

Event type: ◆ SNMP Trap ◇ SunNet Manager Event

Enterprise object id: 1.3.6.1.4.1.45.3.3

Generic trap: Enterprise

Specific trap: 1

Trap expanded name: Fan failure

Schema: AR-HD Calls (aegean)

Figure 18 AR System - SNM Event/Trap Mapper Window

At startup, the process registers as user `SunNet Manager`, so that any translations defined will be legal when the `arsnmd` daemon specifies them.

Like the `arsnmd` daemon, if there is a password for the `SunNet Manager` user, you must use the `-p` command line option to specify the password at startup. If you do not supply the password, you will not be able to create new translations.

For example, to start arsnmui with user `SunNet Manager` having a password of `SNMPass`, enter the following:

```
# <ar_install_dir>/bin/arsnmui -p SNMPass &
```

File

The Action Request System — SunNet Manager Event/Trap Mapper window File menu provides the following options:

- **Save**. Save the changes you have defined to the file `<ar_install_dir>/db/snm.ar`. You can also enter `Ctrl+S`.
- **Exit**. Exit from the tool. You can also enter `Alt+F4`.

Help

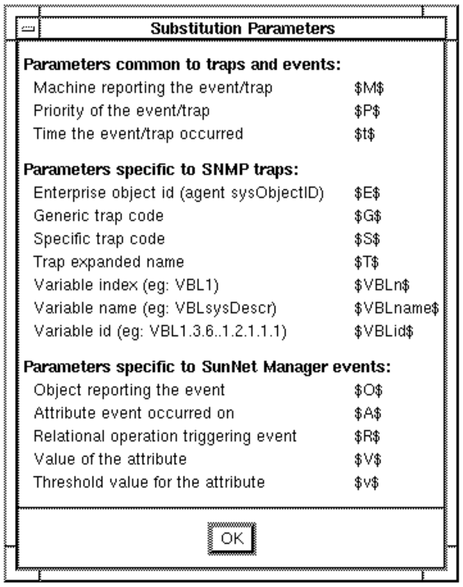
The Help menu provides the following options:

- **Substitution Parameters**. Display the Substitution Parameters window (shown in Figure 19) displaying the parameters available for translating information. You can also enter `Ctrl+P`.
- **On Version**. View information about the tool, such as the version number and on which client machine the tool is running. You can also enter `*`.

Substitution Parameters

You can bring up a display of the parameters you can specify to translate information into the various fields of a new entry by selecting Substitution Parameters from the Help menu. The Substitution Parameters window appears, as shown in Figure 19.

.



Substitution Parameters

Parameters common to traps and events:	
Machine reporting the event/trap	M
Priority of the event/trap	P
Time the event/trap occurred	t
Parameters specific to SNMP traps:	
Enterprise object id (agent sysObjectID)	E
Generic trap code	G
Specific trap code	S
Trap expanded name	T
Variable index (eg: VBL1)	$VBLn$
Variable name (eg: VBLsysDescr)	$VBLname$
Variable id (eg: VBL1.3.6..1.2.1.1.1)	$VBLid$
Parameters specific to SunNet Manager events:	
Object reporting the event	O
Attribute event occurred on	A
Relational operation triggering event	R
Value of the attribute	V
Threshold value for the attribute	v

Figure 19 Substitution Parameters Window

You can use some of the available parameters in both SNMP traps and SNM events; these can be included in any mapping. You should use other values only where appropriate. In most cases, if you use a parameter that doesn't apply to the type of event being translated, the parameter itself will be left in the translation. For `VarBindList` variables, if the specified variable doesn't exist, a null value will be used for the translation.

Table 3 (on the following page) describes all the parameters you can specify to translate information into the various fields of a new entry.

Note – The variable bindings list (`VarBindList`) is a list of variables that are defined to hold specific information about SNMP traps. The `VarBindList` variables in Table 3 begin with the characters **$VBL**.

.

Table 3 Substitution Parameters

Parameter	Value
M	Network address of the agent (machine) reporting the event or trap.
P	The priority of the event or trap.
t	The time that the event or trap occurred. (This is not absolute time. It is 1/100 seconds since the agent was reset.)
Parameters specific to SNMP traps:	
E	The agent's System Object ID.
G	The generic trap code.
S	The enterprise specific trap code.
T	The expanded name of the trap. If the expanded name is not defined, the default is the trap code.
$VBL<*n*>$	Value of the `VarBindList` variable at location **<*n*>** (where ***n*** is the index to the `VarBindList`). Zero (0) is the index to the first variable.
$VBL<*name*>$	Value of the VarBindList variable whose MIB variable name is equal to **<*name*>**.
$VBL<*id*>$	Value of the `VarBindList` variable with an object identifier equal to **<*id*>**.
Parameters specific to SunNet manager events:	
O	The SNM object reporting the event.
A	The attribute of the SNM object that the event occurred on.
R	The relational operation that triggered the event.
V	The value of the attribute that the event occurred on.
v	The threshold value for the attribute the event occurred on.

Note – The SunNet Manager event dispatcher does not simply forward the SNMP trap. It first decodes the trap and modifies values based on how SNM is configured. If it is configured to replace the variable IDs with names, it will replace the VarBindList variable object identifiers with names. If it is not configured in this manner, it keeps the object identifiers as they are. For this reason, the AR System exports both the names and the identifiers of the VarBindList variables.

The simplest way to determine how SNM is configured on your system is to look at the contents of the SNMP trap log file to see whether names or identifiers are logged for VarBindList variables.

Modifying Registrations

The listbox at the top of the Action Request System — SunNet Manager Event/Trap Mapper window (shown in Figure 18 on page 42) contains an ordered list of all of the events already registered. It displays the type of the event (SNMP trap or SNM event) along with any specified conditions.

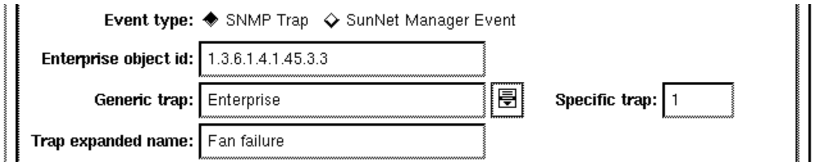
The order of items in the listbox is important as they are processed serially. The first match in the file is chosen regardless of whether or not it is the best match. This is especially important to remember if wild cards are used in the conditions.

You can specify the following options:

- **Insert -** Inserts a new item in the listbox built from the information on the lower half of the screen. You can choose from the selection menu next to the Insert button to specify whether the item should appear at the top of the list, at the bottom, or before or after the currently selected item.
- **Modify -** Modifies the selected item to the new information in the lower half of the window.
- **Delete -** Deletes the currently selected item from the list.

Matching an SNMP Trap

Matching an SNMP trap involves checking the enterprise ID of the object that generated the trap and the type of trap generated. You specify the match parameters by specifying an Event type of SNMP Trap and entering values in the middle of the Action Request System — SunNet Manager Event/Trap Mapper window, as shown in Figure 20.



Event type: ◆ SNMP Trap ◇ SunNet Manager Event

Enterprise object id: 1.3.6.1.4.1.45.3.3

Generic trap: Enterprise

Specific trap: 1

Trap expanded name: Fan failure

Figure 20 Matching an SNMP Trap

- **Event type -** Set to SNMP trap.
- **Enterprise object id -** The enterprise registration ID of the agent you are interested in. If you want to match a class of traps, you can use an asterisk (*) at the end of the ID to specify any trap matching the registration ID to this point.

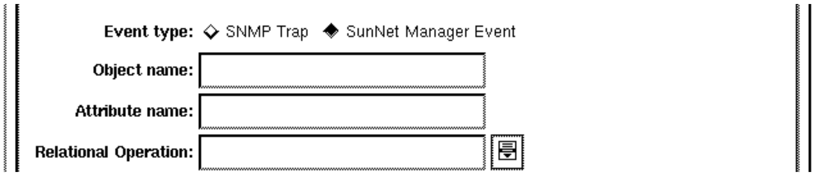
  Note that the order of registration in the listbox is important when using an asterisk (*) to ensure any specific matches (such as 1.3.6.1.4.1.400.5.5) occur before any wild card matches (such as 1.3.6.1.4.1.400.*) or the specific match will always be superceded by the wild card match.
- **Generic trap -** The string identifying the trap type. This string is one of the values included on the menu associated with this field. Select an item from the menu to enter it in the field. If you want to match all types of traps, select All from the menu (or simply leave the field blank). An asterisk (*) is the wild card character.

  Note that SNM allows you to map a trap to a string and that mapping occurs before it is delivered to the AR System daemon. Accordingly, if you have defined mappings for the trap type, you must use the expanded names.
- **Specific trap -** If the value you enter in the Generic trap field is Enterprise, you must enter the enterprise code number for the trap in this field. An asterisk (*) is the wild card character. (If you enter any value other than Enterprise in the Generic trap field, this field is grayed.)

- **Trap expanded name -** If you have not specified a mapping, or you want to remap the trap type string to something more readable or meaningful, enter the string you want used for any trap type (T) substitutions in this field. This field is especially useful for mapping enterprise-specific traps to a friendlier name.

Matching an SNM Event

Matching a SNM event involves checking the object and attribute name of the object that generated the event and the type of event generated. You specify the match parameters by specifying an Event type of SunNet Manager Event and entering values in the middle of the Action Request System — SunNet Manager Event/Trap Mapper window, as shown in Figure 21.



Event type: ◇ SNMP Trap ◆ SunNet Manager Event

Object name:

Attribute name:

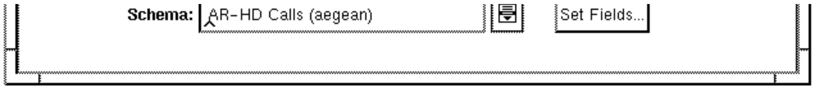
Relational Operation:

Figure 21 Matching a SunNet Manager Event

- **Event type -** Set to SunNet Manager Event.
- **Object name -** Class of object having an interesting threshold. For example, if checking for threshold events on hostperf.data.cpu%, the Object name is data.
- **Attribute name -** Attribute of the object class having an interesting threshold. For example, if checking for threshold events on hostperf.data.cpu%, the Attribute name is cpu%.
- **Relational operation -** The relational operation for the threshold event you are interested in translating. If specified, it must be one of the operations listed on the associated menu. If one is specified, only thresholds for the given operation will be translated. If no value is given, all threshold events will be translated.

Translating an Event to an AR System Entry

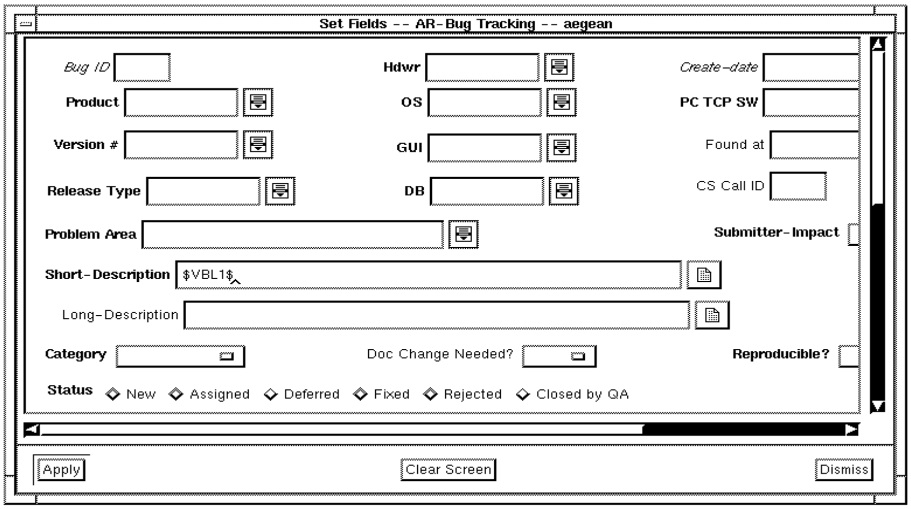
Regardless of whether the event is an SNMP trap or SNM event, you specify the translation to the AR System in the same way, using the final fields on the Action Request System — SunNet Manager Event/Trap Mapper window, as shown in Figure 22.



Schema: AR-HD Calls (aegean)

Figure 22 Translating an SNM Event to an AR System Entry

- **Schema -** The name of the schema to translate into. The menu contains a list of all the accessible schemas. You simply select the menu button to the right of the field then select from the list to enter a schema name in the field.
- **Set Fields -** The Set Fields button brings up a window containing all the fields for the specified schema that the `SunNet Manager` user can set at submit time. An example of the Set Fields window is shown in Figure 23.



Short-Description $VBL1$

Figure 23 Set Fields Window for SNM Event Translation

All the fields set automatically by the AR System and all fields loaded automatically by `arsnmd` are read only. You can specify any text (including parameters) for the other fields. If no value is specified for any of the fields, no value is supplied when the entry is created.

By default, the Short-Description field (field ID=8) is set to the following:

- **SNM event**. Threshold event — `$M$: $O$: $A$`
- **SNMP trap**. SNMP Trap — `$M$: $E$: $T$`

This provides a consistent style and format for traps/events mapped from the SNM event stream.

You can change the value for this field to any value desired. Your value will override the default.

Be aware of the following issues:

- You will not be able to open any Text Edit or Diary Entry windows from the Set Fields window.
- You cannot use AR System keywords in the mapping.
- Query-style menus that reference values on the current screen will produce an error message regarding inconsistent data types, and will not work in the Set Fields window.

Index