



Micro Focus Security ArcSight Connectors
SmartConnector for IBM AIX Audit Syslog
Configuration Guide

June, 2018

Configuration Guide

SmartConnector for IBM AIX Audit Syslog

June, 2018

Copyright © 2016 – 2017; 2018 Micro Focus and its affiliates and licensors.

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus 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. Except as specifically indicated otherwise, a valid license from Micro Focus is 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.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated. Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. UNIX® is a registered trademark of The Open Group.

Revision History

Date	Description
10/17/2017	Added encryption parameters to Global Parameters.
05/15/2017	Created a new zip file for script examples containing individual files for perjoiner-syslogd, perjoiner-syslog4, and streamcmds scripts.
11/30/2016	Updated installation procedure for setting preferred IP address mode. Moved script examples to a separate .txt file for ease of copy and pasting.
06/30/2016	First edition of this Configuration Guide.

SmartConnector for IBM AIX Audit Syslog

This guide provides information for installing the SmartConnector for IBM AIX Audit Syslog and configuring the device for event collection. IBM AIX Audit is supported for collecting auditing events from IBM AIX 7.1.

Product Overview

The purpose of the AIX auditing system is to record instances of access by subjects to objects and to allow detection of any (repeated) attempts to bypass the protection mechanism and any misuses of privileges.

Configuration

Configure AIX Audit

Event Collection

Information collection encompasses logging the selected auditable events. The audit logger is responsible for constructing the complete audit record, consisting of the audit header, which contains information common to all events (such as the name of the event, the user responsible, the time and return status of the event) and the *audit trail*, which contains event-specific information. The audit logger appends each successive record to the kernel audit trail, which can be written in either (or both) BIN and STREAM modes.

Select Audit Events

Auditing lets you detect activities that might compromise the security of your system. When performed by an unauthorized user, the following activities violate system security and are candidates for an audit:

- Engaging in activities in the Trusted Computing Base
- Authenticating users
- Accessing the system
- Changing the configuration of the system
- Circumventing the auditing system
- Initializing the system
- Installing programs
- Modifying accounts
- Transferring information into or out of the system

The audit system does not have a default set of events to be audited. You must select events or event classes according to your needs.

To audit an activity, identify the command or process that initiates the audit event and ensure that the event is listed in the `/etc/security/audit/events` file for your system. Then add the event either to an appropriate class in the `/etc/security/audit/config` file, or to an object stanza in the `/etc/security/audit/objects` file.

See the `/etc/security/audit/events` file on your system for the list of audit events and trail formatting instructions. For a description of how audit event formats are written and used, see the **auditpr** command.

Group into Audit Classes

After you have selected the events to audit, combine similar events into audit classes. These audit classes are defined in the classes stanza of the `/etc/security/audit/config` file. Then assign audit classes to users. Some typical audit classes are:

General

Events that alter the state of the system and change user authentication. Audit attempts to circumvent system access controls.

Objects

Write access to security configuration files.

Kernel

Events in the kernel class are generated by the process management functions of the kernel.

An example of a stanza in the `/etc/security/audit/config` file follows.

```
classes:
    general = USER_SU,PASSWORD_Change,FILE_Unlink,FILE_Link,FILE_Rename
    system = USER_Change,GROUP_Change,USER_Create,GROUP_Create
    init = USER_Login,USER_Logout
```

Assign Audit Events to an Object

Assign the audit events to an object (data or executable file) by adding a stanza for that file to the `/etc/security/audit/objects` file. To get all audit events, specify the ALL class; however, be aware that with this option, a huge amount of data will be generated.

Select an Audit Data Collection Method

The audit data collection method you choose depends upon how you intend to use the audit data. If you need long-term storage of a large amount of data, select BIN collection. If you want to process the data as it is collected, select STREAM collection. If you need both long-term storage and immediate processing, select both methods.



HPE recommends using `streammode=on` and `binmode=off` when using the IBM AIX Audit Syslog SmartConnector.

In the `/etc/security/audit/config` file, configure whether you want to use BIN collection, STREAM collection, or both methods. Use a separate file system for audit data to ensure that audit data does not compete with other data for file space.

To configure STREAM collection:

- 1 Enable the STREAM mode collection by setting `streammode = on` in the Start stanza.
- 2 Edit the Streammode stanza to specify the path to the file containing the streammode processing commands. The default file containing this information is the `/etc/security/audit/streamcmds` file.
- 3 Include the shell commands that process the stream records in an audit pipe in the `/etc/security/audit/streamcmds` file.

Enable the Audit Subsystem

When you have finished making any necessary changes to the configuration files, you can use the `audit start` command to enable the audit subsystem. You can use the `audit shutdown` command to deactivate the audit subsystem.

The auditpr Command

The `auditpr` command reads audit records, in bin or stream format, from standard input and sends formatted records to standard output.

The output format is determined by flags that are selected. If you specify the `-m` flag, a message is displayed before each heading. Use the `-h` flag to change the default fields and the `-v` flag to append an audit trail. The `auditpr` command searches the local `/etc/passwd` file to convert user and group IDs to names.

Values that can be used with the `-h` flag to select fields are as follows:

Value	Description
e	The audit event.
l	The user's login name.
R	The audit status.
t	The time the record was written.
c	The command name.
r	The real user name.
p	The process ID.
i	The IDs or the names of roles of the audited process.
E	The effective privilege.
S	The effective sensitivity label (SL).
I	The effective integrity label (TL).
W	The workload partition name.
P	The ID of the parent process.
T	The kernel thread ID (local to the process; different process can contain threads with the same thread ID).

Value	Description
h	The name of the host that generated the audit record. If there is no CPU ID in the audit record, the value none is used. If there is no matching entry for the CPU ID in the audit record, the 16-character value for the CPU ID is used instead.

The **e**, **I**, **R**, **t**, and **c** flags are used by default.



For more information on `auditpr` commands, see https://www.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.cmds1/auditpr.htm.

Examples

Sample Event File

An example of the `/etc/security/audit/event` file:

```
[#]/etc/security/audit> cat events
....
auditpr:

...other rows precede

*kernel proc events

*      fork()
      PROC_Create = printf "forked child process %d"

*      exit()
      PROC_Delete = printf "exited child process %d"

*      exec()
      PROC_Execute = printf "euid: %d egid: %d epriv: %x:%x name %s"

... other rows follow
```

For examples of audit trails, see the `/etc/security/audit/events` file where the audit trail formats are defined.

Example of auditpr Command

```
[#][/] /usr/sbin/audit pr -v < audit/trail
event          login      status      time          command
-----
FS_Chdir       root      OK          Tue Oct 05 12:58:26 2004 ksh
FILE_Unlink    root      OK          Tue Oct 05 12:59:03 2004 vi
FILE_Unlink    root      OK          Tue Oct 05 12:59:12 2004 vi
FS_Chdir       root      OK          Tue Oct 05 12:59:34 2004 ksh
FS_Chdir       root      OK          Tue Oct 05 12:59:37 2004 ksh
```

```

FILE_Unlink      root      OK      Tue Oct 05 12:59:40 2004 vi
FILE_Unlink      root      OK      Tue Oct 05 12:59:59 2004 vi
CRON_Start       root      OK      Tue Oct 05 13:00:00 2004 cron
FS_Chdir         root      OK      Tue Oct 05 13:00:00 2004 cron
FILE_Unlink      root      OK      Tue Oct 05 13:00:02 2004 vi
FILE_Unlink      root      OK      Tue Oct 05 13:00:04 2004 vi
FILE_Unlink      root      OK      Tue Oct 05 13:02:38 2004 vi
FILE_Unlink      root      OK      Tue Oct 05 13:02:44 2004 vi
FILE_Unlink      root      OK      Tue Oct 05 13:02:44 2004 vi
TCPIP_connect    root      OK      Tue Oct 05 13:20:15 2004 telnetd
FILE_Write       root      OK      Tue Oct 05 13:20:15 2004 telnetd

```

Example of Config File for BIN Mode

```

[#][/etc/security/audit]> head -20 config

start:
    binmode = on
    streammode = on

bin:
    trail = /audit/trail
    bin1 = /audit/bin1
    bin2 = /audit/bin2
    binsize = 10240
    cmds = /etc/security/audit/bincmds
    freespace = 65536

...
[#][/etc/security/audit]> cat /etc/audit/audit/bincmds
/usr/sbin/auditcat -p -o $trail $bin
[p630n02][/etc/security/audit]>

```

Example of Config File for STREAM Mode

```

[#][/etc/security/audit]> cat config

start:
    binmode = on
    streammode = on

stream:
    cmds = /etc/security/audit/streamcmds

...
[#][/etc/security/audit]> cat /etc/security/audit/streamcmds
/usr/sbin/auditstream | auditpr -v > /audit/stream.out &
[#][/ETC/SECURITY/AUDIT]>

```

Example of a Generic Audit Log Scenario

This example was derived from the *AIX Security Guide* in the *Setting Up Auditing* chapter in the section called *Generating a Generic Audit Log*. See https://www.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.security/generic_audit_log.htm for details.

In this example, assume that a SYSADMIN wants to use the audit subsystem to monitor a large multi-user server system. No direct integration into an IDS is performed, all audit records will be inspected manually for irregularities. Only a few essential audit events are recorded, to keep the amount of generated data to a manageable size.

The audit events that are considered for event detection are:

FILE_WRITE	We want to know about file writes to configuration files, so this event will be used with all files in the /etc tree.
PROC_SetUserIDs	All changes of user ids.
AUD_Bin_Def	Audit bin configuration.
USER_SU	The su command.
PASSWORD_Change	The passwd command.
AUD_Lost_Rec	Notification in case there where lost records.
CRON_JobAdd	New cron jobs.
AT_JobAdd	New at jobs.
USER_Login	All logins.
PORT_Locked	All locks on terminals because of too many invalid attempts.

The following is an example of how to generate a generic audit log:

- 1 Set up a list of critical files to be monitored for changes, such as all files in **/etc**, and configure them for **FILE_Write** events in the **objects** file as follows:

```
find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n", $1)}' >>
/etc/security/audit/objects
```

- 2 Use the **auditcat** command to set up BIN mode auditing. The **/etc/security/audit/bincmds** file is similar to the following:

```
/usr/sbin/auditcat -p -o $trail $bin
```

- 3 Edit the **/etc/security/audit/config** file and add a class for the events in which we are interested. List all existing users and specify the custom class for them.

```
start:
    binmode = on
    streammode = off

bin:
    cmds = /etc/security/audit/bincmds
    trail = /audit/trail
```



```

bin1 = /audit/bin1
bin2 = /audit/bin2
binsize = 100000
freespace = 100000

classes:
    custom = FILE_Write,PROC_SetUser,AUD_Bin_Def,AUD_Lost_Rec,
            USER_SU,PASSWORD_Change,CRON_JobAdd,AT_JobAdd,USER_Login,
            PORT_Locked

users:
    root = custom
    afx = custom
    ...

```

- 4 Add the `custom` audit class to the `/usr/lib/security/mkuser.default` file, so that new IDs will automatically have the correct audit call associated:

```

user:
    auditclasses = custom
    pgrp = staff
    groups = staff
    shell = /usr/bin/ksh
    home = /home/$USER

```

- 5 Create a new file system named `/audit` by using SMIT or the `crfs` command. The file system should be large enough to hold the two bins and a large audit trail.
- 6 Run the `audit start` command option and examine the `/audit` file. You should see the two bin files and an empty `trail` file initially. After you have used the system for a while, you should have audit records in the `trail` file that can be read with:

```
auditpr -hhhelpPRrTc -v | more
```

This example uses only a few events. To see all events, you could specify the classname `ALL` for all users. This action will generate large amounts of data. You might want to add all events related to user changes and privilege changes to your `custom` class.

Example of Real-Time File Modification Monitoring

The following example can be used to monitor file access to critical files in real time:

- 1 Set up a list of critical files to be monitored for changes; for example, all files in `/etc`, and configure them for `FILE_Write` events in the `objects` file.

```
find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >>
/etc/security/audit/objects
```

- 2 Set up stream auditing to list all file writes. (This example lists all file writes to the console, but in using the ArcSight SmartConnector in a production environment, you would want to have a

backend that sends the events into an Intrusion Detection System.) The **/etc/security/audit/streamcmds** file is similar to the following:

```
/usr/sbin/auditstream | /usr/sbin/auditselect -e "event == FILE_Write" |
auditpr -hhhelpPRtTc -v > /dev/console &
```

- 3 Set up STREAM mode auditing in **/etc/security/audit/config**; add a class for the file write events and configure all users that should be audited with that class:

```
start:
    binmode = off
    streammode = on

stream:
    cmds = /etc/security/audit/streamcmds

classes:
    filemon = FILE_Write

users:
    root = filemon
    afx = filemon
    ...
```

- 4 Now run **audit start**. All FILE_Write events are displayed on the console.



When the audit start or audit shutdown command is executed, the configuration information is reset and the audit logs are flushed to the streams. When this happens, the SmartConnector must be restarted.

AIX Configuration Files

AIX configuration files you may need to access include:

File	Description
/usr/sbin/auditselect	Specifies the path of the auditselect command.
/etc/rc	Contains the system initialization commands.
/etc/security/audit/config	Contains audit system configuration information.
/etc/security/audit/events	Contains the audit events of the system.
/etc/security/audit/objects	Contains audit events for audit objects (files).
/etc/security/audit/bincmds	Contains auditbin backend commands.
/etc/security/audit/streamcmds	Contains auditstream commands.

See the [AIX 7.1 Security](#) manual for specific information about configuring AIX auditing for your AIX version.

Select a Method to Send AIX Audit Messages Using Syslog

AIX Audit messages are multiline and are not supported by syslog. The following instructions allow the multiline messages to be reassembled to be forwarded and understood by the IBM AIX Audit Syslog SmartConnector in one of three ways:

- Solution 1: Send the messages to the local syslog daemon using a UNIX socket and configure the daemon to forward the messages to the SmartConnector.
- Solution 2: Send the messages directly to the SmartConnector using syslog UDP.
- Solution 3: Send the messages using syslog UDP to the local syslog daemon and configure the daemon to forward the messages to the SmartConnector.

To implement your preferred solution:

- Create the three perl scripts as shown in the "Create Three Perl Scripts" section.
- Follow the instructions in the "Configure AIX to Issue Audit Messages in Stream" section.

Caveats about Implementing Solutions

- The Solution 3 setup cannot be used with stock AIX syslogd. Instructions vary depending on syslog daemon installed. In this case, you have three choices: On the UDP receiving port, increase the UDP buffers available in the system to avoid message loss; use TCP to forward the log to the SmartConnector; or, configure a cache size on the TCP connection that provides guaranteed delivery.
- Only solutions 1 and 3 support keeping a local copy of the log on the issuing server. Solution 3 is not supported if you are using stock AIX syslogd.
- Refer to the Manage Prefixes section under Additional Configuration (after installing the core connector software) to learn how to manage the parsing of custom AIX-specific forwarding prefixes and remove the "forwarding message" phrase.
- It is important to note the difference between using UDP and Raw TCP. To ensure data integrity, Raw TCP should be used. For better performance, use UPD.
- The example scripts provided work, but cannot be guaranteed to function by HPE ArcSight on every implementation of an AIX Audit system.

Create Three Perl Scripts

Create the following three perl scripts to implement the solutions that allow multiline messages to be changed and used by the IBM AIX Audit Syslog SmartConnector. For examples of these scripts, see "SmartConnector for IBM AIX Audit Syslog Script Examples" at <https://www.protect724.hpe.com/docs/DOC-14626>.

If you downloaded the SmartConnector configuration guide zip file from the SSO site, click [here](#) for the script examples.

- 1 Create a perl script named `/etc/security/audit/streamcmds` (or `streamcmds.orig` if you don't want to overwrite the default `streamcmds` file) as shown in the `streamcmds` example in the `IBMAIXAuditSyslogScriptExamples.zip` file.
- 2 Create a perl script named `/etc/security/audit/perljoiner-syslogd` as shown in the `perljoiner-syslogd` example in the `IBMAIXAuditSyslogScriptExamples.zip` file.
- 3 Create a perl script named `/etc/security/audit/perljoiner-syslogr` as shown in the `perljoiner-syslogr` example in the `IBMAIXAuditSyslogScriptExamples.zip` file.

Configure AIX to Issue Audit Messages in Stream

- 1 Edit the `/etc/security/audit/config` file.

```
start:
    binmode = off
    streammode = on

stream:
    streamcompact = off
    cmds = /etc/security/audit/streamcmds
```

- 2 Install the SmartConnector.
- 3 Implement one of the three solutions mentioned earlier, as shown in the sections below.
- 4 (Optional) Remove the phrase message forwarded as shown in the "Additional Configuration" section.

Implement Solution 1

This solution allows you to send the messages to the local syslog daemon using UNIX socket and configure the daemon to forward the messages to the SmartConnector.

- 1 Go to the `/etc/security/audit/streamcmds` file.
- 2 Include the line that uses `perljoiner-syslogd` and comment out the line that uses `perljoiner-syslogr`

```
#Use this to send to the local syslog daemon
/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogd &
#Use this to send to a remote destination such as smartconnector
#/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogr &
```

- 3 Modify the syslog daemon to redirect the audit log to the SmartConnector.



By default, logs will get to syslog daemon with facility-priority set as `local0.info` and with `pgmname` set as `"auditpr"`. `syslogd` only supports UDP implicit port 514, `rsyslog` and `syslog-ng` support TCP and UDP and a specific port can be supplied.

```
#To forward using udp with implicit port 514
local0.info @ipSmartconnector
```

```
#To forward using udp with specific port
local0.info @ipSmartconnector:port
```

```
#To forward using tcp with specific port
local0.info @@ipSmartconnector:port
```



If you use stock AIX syslogd, verify that the `-n` switch is not configured when the daemon is started. This switch instructs syslogd to add the hostname inside the syslog message. The hostname presence in the syslog message is required by the SmartConnector.

- 4 Edit `/etc/security/audit/config/perljoiner-syslogd` to adjust the following two lines to values for your environment:

```
my $maxretry=20;
my $delaybetweenretry=30;
```

Implement Solution 2

This solution allows you to send the messages directly to the SmartConnector using syslog UDP.

- 1 Go to the `/etc/security/audit/streamcmds` file.
- 2 Include the line that uses `perljoiner-syslogr` and comment out the line that uses `perljoiner-syslogd`

```
#Use this to send to the local syslog daemon
#/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogd &
#Use this to send to a remote destination such as smartconnector
/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogr &
```

- 3 Edit `/etc/security/audit/perljoiner-syslogr` to change the destination address to the following address and port:

```
SyslogHost => 'ipSmartconnector'
SyslogPort => 'port'
```

Implement Solution 3

This solution allows you to send the messages using syslog UDP to the local syslog daemon and configure the daemon to forward the messages to the SmartConnector.

- 1 Go to the `/etc/security/audit/streamcmds` file.
- 2 Include the line that uses `perljoiner-syslogr` and comment out the line that uses `perljoiner-syslogd`.

```
#Use this to send to the local syslog daemon
#/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogd &
#Use this to send to a remote destination such as smartconnector
/usr/sbin/auditstream | /usr/sbin/auditpr -v -t0 -h e,l,R,t,c,p,P | perl
/etc/security/audit/perljoiner-syslogr &
```

- 3 Edit `/etc/security/audit/perljoiner-syslogr` to change the destination address to the following address and port:

```
SyslogHost => '127.0.0.1'
SyslogPort => '11514'
```

- 4 Modify syslog daemon to listen on UDP 11514 and redirect audit log to the SmartConnector.

Modifying the Daemon to Listen on UDP and Redirecting the Audit Log

Redirection instructions vary depending on your installed syslog daemon. Below are sample configurations for rsyslog and syslog-ng.

To avoid message loss on the AIX syslog daemon:

- Use TCP to forward the log to the SmartConnector instead of UDP.
- Increase the UDP buffers available on the UDP receiving port.
- Configure guaranteed delivery on the TCP connection to the SmartConnector.

Sample rsyslog configuration

```
# Provides UDP syslog reception on port 11514
    $ModLoad imudp
    $UDPServerRun 11514
#To forward using TCP
    local0.info @@ipSmartconnector:port
```

Sample syslog-ng configuration

```
source s_udplocal { tcp(ip(127.0.0.1) port(11514)
so_rcvbuf(2097152)); };
destination remote_smart { tcp("ipSmartconnector" port(port));
log { source(s_udplocal); destination(remote_smart); };
```

Configure the Syslog SmartConnectors

The three ArcSight Syslog SmartConnectors are:

- Syslog Daemon
- Syslog Pipe
- Syslog File

The Syslog Daemon SmartConnector

The Syslog Daemon SmartConnector is a syslogd-compatible daemon designed to work in operating systems that have no syslog daemon in their default configuration, such as Microsoft Windows. The SmartConnector for Syslog Daemon implements a UDP receiver on port 514 (configurable) by default that can be used to receive syslog events. Use of the TCP protocol or a different port can be configured manually.

If you are using the SmartConnector for Syslog Daemon, simply start the connector, either as a service or as a process, to start receiving events; no further configuration is needed.



Messages longer than 1024 bytes may be split into multiple messages on syslog daemon; no such restriction exists on syslog file or pipe.

The Syslog Pipe and File SmartConnectors

When a syslog daemon is already in place and configured to receive syslog messages, an extra line in the syslog configuration file (`rsyslog.conf`) can be added to write the events to either a **file** or a system **pipe** and the ArcSight SmartConnector can be configured to read the events from it. **In this scenario, the ArcSight SmartConnector runs on the same machine as the syslog daemon.**

The **Syslog Pipe** SmartConnector is designed to work with an existing syslog daemon. This SmartConnector is especially useful when storage is a factor. In this case, syslogd is configured to write to a named pipe, and the Syslog Pipe SmartConnector reads from it to receive events.

The **Syslog File** SmartConnector is similar to the Pipe SmartConnector; however, this SmartConnector monitors events written to a syslog file (such as `messages.log`) rather than to a system pipe.

Configure the Syslog Pipe or File SmartConnector

This section provides information about how to set up your existing syslog infrastructure to send events to the ArcSight Syslog Pipe or File SmartConnector.

The standard UNIX implementation of a syslog daemon reads the configuration parameters from the `/etc/rsyslog.conf` file, which contains specific details about which events to write to files, write to pipes, or send to another host. First, create a pipe or a file; then modify the `/etc/rsyslog.conf` file to send events to it.

For syslog pipe:

- 1 Create a pipe by executing the following command:

```
mkfifo /var/tmp/syspipe
```

- 2 Add the following line to your `/etc/rsyslog.conf` file:

```
*.debug /var/tmp/syspipe
```

or

```
*.debug |/var/tmp/syspipe
```

depending on your operating system.

- 3 After you have modified the file, restart the syslog daemon either by executing the scripts `/etc/init.d/syslogd stop` and `/etc/init.d/syslogd start`, or by sending a `configuration restart` signal.

On RedHat Linux, you would execute:

```
service syslog restart
```

On Solaris, you would execute:

```
kill -HUP `cat /var/run/syslog.pid`
```

This command forces the syslog daemon to reload the configuration and start writing to the pipe you just created.

For syslog file:

Create a file or use the default for the file into which log messages are to be written.

After editing the `/etc/rsyslog.conf` file, be sure to restart the syslog daemon as described above.

When you follow the SmartConnector Installation Wizard, you will be prompted for the absolute path to the syslog file or pipe you created.

Install the SmartConnector

The following sections provide instructions for installing and configuring your selected SmartConnector.

Syslog Installation

Install this SmartConnector (on the syslog server or servers identified in the *Configuration* section) using the SmartConnector Installation Wizard appropriate for your operating system. The wizard will guide you through the installation process. When prompted, select one of the following **Syslog** connectors (see *Configure the Syslog SmartConnectors* in this guide for more information):

- Syslog Daemon
- Syslog Pipe
- Syslog File

Because all syslog SmartConnectors are sub-connectors of the main syslog SmartConnector, the name of the specific syslog SmartConnector you are installing is not required during installation.

The syslog daemon connector by default listens on port 514 (configurable) for UDP syslog events; you can configure the port number or use of the TCP protocol manually. The syslog pipe and syslog file connectors read events from a system pipe or file, respectively. Select the one that best fits your syslog infrastructure setup.

Prepare to Install Connector

Before you install any SmartConnectors, make sure that the ArcSight products with which the connectors will communicate have already been installed correctly (such as ArcSight ESM or ArcSight Logger).

For complete product information, read the *Administrator's Guide* as well as the *Installation and Configuration* guide for your ArcSight product before installing a new SmartConnector. If you are adding a connector to the ArcSight Management Center, see the *ArcSight Management Center Administrator's Guide* for instructions, and start the installation procedure at "Set Global Parameters (optional)" or "Select Connector and Add Parameter Information."

Before installing the SmartConnector, be sure the following are available:

- Local access to the machine where the SmartConnector is to be installed
- Administrator passwords

Install Core Software

Unless specified otherwise at the beginning of this guide, this SmartConnector can be installed on all ArcSight supported platforms; for the complete list, see the *SmartConnector Product and Platform Support* document, available from the Micro Focus SSO and Protect 724 sites.

- 1 Download the SmartConnector executable for your operating system from the Micro Focus SSO site.
- 2 Start the SmartConnector installation and configuration wizard by running the executable.

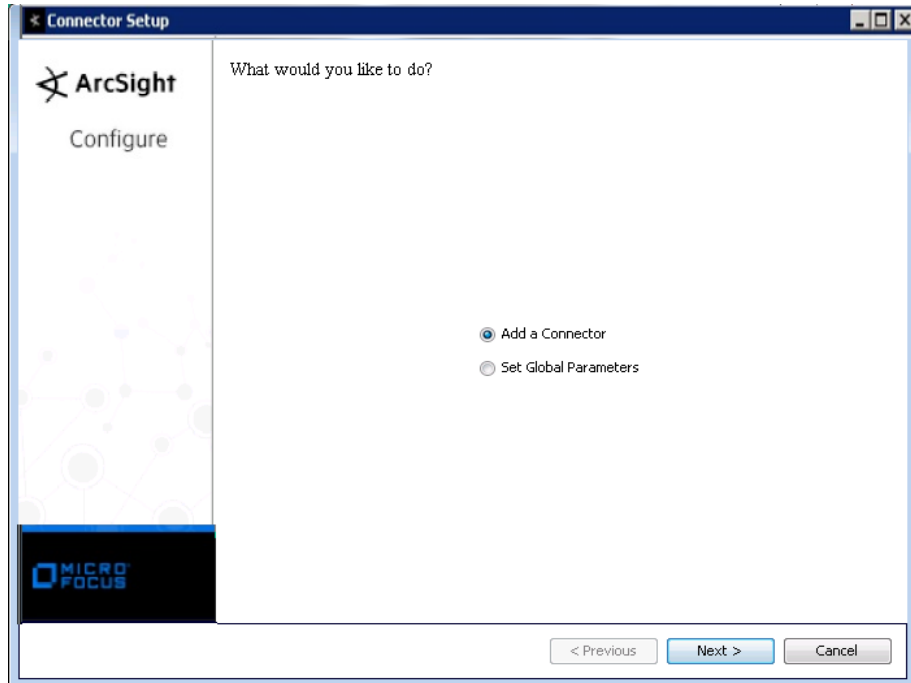


When installing a syslog daemon SmartConnector in a UNIX environment, run the executable as 'root' user.

Follow the wizard through the following folder selection tasks and installation of the core connector software:

Introduction
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...

- 3 When the installation of SmartConnector core component software is finished, the following window is displayed:



Set Global Parameters (optional)

If you choose to perform any of the operations shown in the following table, do so before adding your connector. You can set the following parameters:

Parameter	Setting
FIPS mode	Select 'Enabled' to enable FIPS compliant mode. To enable FIPS Suite B Mode, see the SmartConnector User Guide under "Modifying Connector Parameters" for instructions. Initially, this value is set to 'Disabled'.
Remote Management	Select 'Enabled' to enable remote management from ArcSight Management Center. When queried by the remote management device, the values you specify here for enabling remote management and the port number will be used. Initially, this value is set to 'Disabled'.
Remote Management Listener Port	The remote management device will listen to the port specified in this field. The default port number is 9001.
Preferred IP Version	When both IPv4 and IPv6 IP addresses are available for the local host (the machine on which the connector is installed), you can choose which version is preferred. Otherwise, you will see only one selection. The initial setting is IPv4.

The following parameters should be configured only if you are using Micro Focus SecureData solutions to provide encryption. See the *Micro Focus SecureData Architecture Guide* for more information.

Parameter	Setting
Format Preserving Encryption	Data leaving the connector machine to a specified destination can be encrypted by selecting 'Enabled' to encrypt the fields identified in 'Event Fields to Encrypt' before forwarding events. If encryption is enabled, it cannot be disabled. Changing any of the encryption parameters again will require a fresh installation of the connector.
Format Preserving Policy URL	Enter the URL where the Micro Focus SecureData Server is installed.
Proxy Server (https)	Enter the proxy host for https connection if any proxy is enabled for this machine.
Proxy Port	Enter the proxy port for https connection if any proxy is enabled for this machine.

Parameter	Setting
Format Preserving Identity	The Micro Focus SecureData client software allows client applications to protect and access data based on key names. This key name is referred to as the identity. Enter the user identity configured for Micro Focus SecureData.
Format Preserving Secret	Enter the secret configured for Micro Focus SecureData to use for encryption.
Event Fields to Encrypt	Recommended fields for encryption are listed; delete any fields you do not want encrypted and add any string or numeric fields you want encrypted. Encrypting more fields can affect performance, with 20 fields being the maximum recommended. Also, because encryption changes the value, rules or categorization could also be affected. Once encryption is enabled, the list of event fields cannot be edited.

After making your selections, click **Next**. A summary screen is displayed. Review the summary of your selections and click **Next**. Click **Continue** to return to proceed with "Add a Connector" window. Continue the installation procedure with "Select Connector and Add Parameter Information."

Select Connector and Add Parameter Information

- 1 Select **Add a Connector** and click **Next**. If applicable, you can enable FIPS mode and enable remote management later in the wizard after SmartConnector configuration.
- 2 Select **Syslog Daemon, Syslog File, or Syslog Pipe** and click **Next**.
- 3 Enter the required SmartConnector parameters to configure the SmartConnector, then click **Next**.

Syslog Daemon Parameters	<i>Network port</i>	The SmartConnector for Syslog Daemon listens for syslog events from this port.
	<i>IP Address</i>	The SmartConnector for Syslog Daemon listens for syslog events only from this IP address (accept the default (ALL) to bind to all available IP addresses).
	<i>Protocol</i>	The SmartConnector for Syslog Daemon uses the selected protocol (UDP or Raw TCP) to receive incoming messages.
	<i>Forwarder</i>	Change this parameter to 'true' only if the events being processed are coming from another SmartConnector sending to a CEF Syslog destination, and that destination also has CEF forwarder mode enabled. That allows attributes of the original connector to be retained in the original agent fields.
Syslog Pipe Parameter	<i>Pipe Absolute Path Name</i>	Absolute path to the pipe, or accept the default: <code>/var/tmp/syspipe</code>
Syslog File Parameters	<i>File Absolute Path Name</i>	<p>Enter the full path name for the file from which this connector will read events or accept the default: <code>\var\adm\messages</code> (Solaris) or <code>\var\log\messages</code> (Linux).</p> <p>A wildcard pattern can be used in the file name; however, in realtime mode, rotation can occur only if the file is over-written or removed from the folder. Realtime processing mode assumes following external rotation.</p> <p>For date format log rotation, the device writes to 'filename.timestamp.log' on a daily basis. At a specified time, the device creates a new daily log and begins to write to it. The connector detects the new log and terminates the reader thread to the previous log after processing is complete. The connector then creates a new reader thread to the new 'filename.timestamp.log' and begins processing that file. To enable this log rotation, use a date format in the file name as shown in the following example:</p> <pre>filename'yyyy-MM-dd'.log;</pre>

For index log rotation, the device writes to indexed files - 'filename.log.001', 'filename.log.002', 'filename.log.003', and so on. At startup, the connector processes the log with highest index. When the device creates a log with a greater index, the connector terminates the reader thread to the previous log after processing completes, creates a thread to the new log, and begins processing that log. To enable this log rotation, use an index format, as shown in the following example:

```
filename'%d,1,99,true'.log;
```

Specifying 'true' indicates that it is allowed for the index to be skipped; for example, if 5 appears before 4, processing proceeds with 5 and will not read 4, even if 4 appears later. Use of 'true' is optional.

<i>Reading Events Real Time or Batch</i>	Specify whether file is to be read in batch or realtime mode. For batch mode, all files are read from the beginning. The 'Action Upon Reaching EOF' and 'File Extension if Rename Action' parameters apply for batch mode only.
<i>Action Upon Reaching EOF</i>	For batch mode, specify 'None', 'Rename', or 'Delete' as the action to be performed to the file when the connector has finished reading and reaches end of file (EOF). For realtime mode, leave the default value of 'None' for this parameter.
<i>File Extension If Rename Action</i>	For batch mode, specify the extension to be added to the file name if the action upon EOF is 'Rename' or accept the default value of '.processed'.

Select a Destination

- 1 The next window asks for the destination type; select a destination and click **Next**. For information about the destinations listed, see the *ArcSight SmartConnector User Guide*.
- 2 Enter values for the destination. For the ArcSight Manager destination, the values you enter for **User** and **Password** should be the same ArcSight user name and password you created during the ArcSight Manager installation. Click **Next**.
- 3 Enter a name for the SmartConnector and provide other information identifying the connector's use in your environment. Click **Next**. The connector starts the registration process.
- 4 If you have selected ArcSight Manager as the destination, the certificate import window for the ArcSight Manager is displayed. Select **Import the certificate to the connector from destination** and click **Next**. (If you select **Do not import the certificate to the connector from destination**, the connector installation will end.) The certificate is imported and the **Add connector Summary** window is displayed.

Complete Installation and Configuration

- 1 Review the **Add Connector Summary** and click **Next**. If the summary is incorrect, click **Previous** to make changes.
- 2 The wizard now prompts you to choose whether you want to run the SmartConnector as a stand-alone process or as a service. If you choose to run the connector as a stand-alone process, select **Leave as a standalone application**, click **Next**, and continue with step 5.
- 3 If you chose to run the connector as a service, with **Install as a service** selected, click **Next**. The wizard prompts you to define service parameters. Enter values for **Service Internal Name** and **Service Display Name** and select **Yes** or **No** for **Start the service automatically**. The **Install Service Summary** window is displayed when you click **Next**.

- 4 Click **Next** on the summary window.
- 5 To complete the installation, choose **Exit** and Click **Next**.

For instructions about upgrading the connector or modifying parameters, see the *SmartConnector User Guide*.

Additional Configuration

Manage Prefixes

You can manage parsing of custom AIX-specific forwarding prefixes by adding properties to the `agent.properties` file (located at `$ARCSIGHT_HOME\current\user\agent`). These properties can be found in the `agent.default.properties` file (located at: `$ARCSIGHT_HOME\current\config\agent`).

The following property controls whether custom AIX-specific **forwarding prefixes** and **facility.priority** portions of the headers are removed. This property is disabled (set to 'false') by default. To remove the "forwarding message" phrase, change the value to 'true', as shown below. Note that setting this value to 'true' may cause some performance degradation.

```
syslog.aix.enabled=true
```

The following property is used to strip out the prefix that AIX adds when it forwards a syslog message to another host:

```
syslog.aix.forwarded.prefixes=Message forwarded from,Forwarded from
syslog.aix.forwarded.prefixes.delimiter=,
```

Run the SmartConnector

SmartConnectors can be installed and run in stand-alone mode, on Windows platforms as a Windows service, or on UNIX platforms as a UNIX daemon, depending upon the platform supported. On Windows platforms, SmartConnectors also can be run using shortcuts and optional Start menu entries.

If the connector is installed in stand-alone mode, it must be started manually and is not automatically active when a host is restarted. If installed as a service or daemon, the connector runs automatically when the host is restarted. For information about connectors running as services or daemons, see the *ArcSight SmartConnector User Guide*.

To run all SmartConnectors installed in stand-alone mode on a particular host, open a command window, go to `$ARCSIGHT_HOME\current\bin` and run: `arcsight connectors`

To view the SmartConnector log, read the file `$ARCSIGHT_HOME\current\logs\agent.log`; to stop all SmartConnectors, enter `Ctrl+C` in the command window.

Device Event Mapping to ArcSight Fields

The following section lists the mappings of ArcSight data fields to the device's specific event definitions. See the *ArcSight Console User's Guide* for more information about the ArcSight data fields.

IBM AIX Audit Event Mappings to ArcSight Fields

ArcSight ESM Field	Device-Specific Field
Agent (Connector) Severity	Medium = FAIL, Low = OK
Device Action	status
Device Custom Number 1	File Descriptor
Device Custom Number 2	Parent PID
Device Custom Number 3	Physical Volume Index
Device Custom String 1	ACL
Device Custom String 2	Group
Device Custom String 3	Owner
Device Custom String 4	Reason or Error Code
Device Custom String 5	PCL
Device Custom String 6	Volume Group ID
Device Event Class ID	event
Device Product	'AIX Audit'
Device Receipt Time	time
Device Severity	status
Device Vendor	'IBM'
Event Outcome	status
Message	message
Name	event
Source Process ID	process
Source Service Name	command
Source User Name	login
