



Micro Focus Security ArcSight Connectors

SmartConnector User Guide

Document Release Date: July 24, 2019

Software Release Date: July 24, 2019

Legal Notices

Copyright Notice

© Copyright 2000-2019 Micro Focus or one of its affiliates

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

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

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

US. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, "commercial computer software" is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation ("FAR") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("DOD"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement ("DFARS") and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

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.

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://software.support.softwaregrp.com/support-contact-information
Support Web Site	https://software.support.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Document Revision History

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.

To check for recent updates or to verify that you are using the most recent edition of a document, go to [ArcSight Product Documentation Community on the Micro Focus Security Community](#).

Document Changes

Date	Product Version	Description

Contents

Chapter 1: About Connectors	11
Data Encryption	13
Connector Data Collection	14
Mapping to Vendor Events	14
Filter and Aggregate Events	15
Connector Types	15
File Connectors	16
Database Connectors	17
API Connectors	17
SNMP Connectors	18
Microsoft Windows Event Log Connectors	18
Syslog Connectors	19
Scanner Connectors	20
FlexConnectors	21
Model Import Connectors	21
Other Connectors	22
Connectors that Use Multiple Mechanisms	22
Connectors that Use TCP in Special Formats	22
Management Center	22
ArcSight Logger	23
Instant Connector Deployment	23
Connectors in Transformation Hub	23
Chapter 2: Planning for Deployment	25
Deployment Overview	25
Supported Platforms	25
Deployment Scenarios	26
Deployment Scenario One	26
Deployment Scenario Two	27
Deployment Scenario Three	28
Deployment Scenario Four	28
Estimating Storage Requirements	28

Understanding ArcSight Turbo Modes	29
Chapter 3: Installing Connectors	30
Installing the Connector from an Executable	30
Installing Connectors from the Command Line	31
Installing Connectors in Silent Mode	31
Uninstalling a Connector	35
Upgrading Connectors from ESM	35
Upgrade Notes	36
Local Upgrade	37
Remote Upgrade from ESM	37
Running Connectors	38
Run Connectors in Standalone Mode	38
Run Connectors as a Windows Service	38
Run Connectors as a UNIX Daemon	39
User Privileges When Installing (UNIX only)	39
When Running As a Service	39
When Running in Standalone Mode	41
Verifying that a Connector is Running on ESM	42
Uninstalling a Connector	42
Working with the Windows Hosts Table	43
Manually Entering Table Parameter Values	43
Importing and Exporting CSV Files	44
Chapter 4: Configuring Connectors	46
Modifying Connectors	46
Modify Connector	46
Modify Connector Parameters	47
Add, Modify, or Remove Destinations	47
Modify Destination Parameters	48
Modify Destination Settings	48
Reregister Destination	49
Add a Failover Destination	50
Install as a Service	50
Set Global Parameters	51
Additional Configuration	54
Enabling FIPS Suite B Mode	54

Specifying the IP Address on Devices with Multiple Network Interfaces	54
Getting the Name of the Network Interface:	55
On an Linux Platform:	55
On a Windows Platform	55
Lowering Network Bandwidth When Sending Events to ESM	56
Defining Default and Alternate Configurations from the ArcSight Console	56
Customized Events Filtering	58
Feature Usage	58
Java Regex Patterns	59
Get Status	59
Examples of Patterns	60
Log Messages in agent.log	62
Connector Network and Zone Configuration	63
Global Event ID Feature	65
Global Event ID Parameters:	65
Unique Generator ID	65
Updating a Unique Generator ID	65
Enabling the Global Event ID Feature	65
Fresh Standalone Installation	66
ArcMC Managed Connectors	66
Upgrading the Connector	66
Local Upgrade	66
Remote Upgrade from ESM	66
Remote Upgrade from ArcMC	66
Chapter 5: Connectors with Management Center	68
Managing Connectors on Management Center	68
Local (on-board) Connectors	68
Remote Management Center Connectors	69
Software-Based Connectors	69
Login Credentials for Software-Based Connector Remote Management	70
Choosing a Deployment Scenario	70
ArcSight Logger	70
ArcSight ESM	71
ESM and Logger	71
Chapter 6: Connector Destinations Overview	72
Connector Destinations	72
ArcSight Manager (encrypted)	72

ArcSight Logger SmartMessage (encrypted)	73
ArcSight Logger SmartMessage Pool (encrypted)	73
CEF File	73
Transformation Hub	73
CEF Syslog	73
CEF Encrypted Syslog (UDP)	74
CSV File	74
Raw Syslog	74
Appendix 6: Add Destinations	74
Failover Destinations	75
Chapter 7: ArcSight Manager Destination	76
ArcSight Manager (encrypted)	76
Chapter 8: Configuring Destination Settings	79
Managing SmartConnector Filter Conditions	89
Chapter 9: ArcSight Logger SmartMessage (encrypted) Destination	91
Sending Events from Logger to a Manager	91
Sending Events to Logger	92
Sending Events to Both Logger and a Manager	94
Forwarding Events from ESM to Logger	96
Defining Connector Settings in Logger	97
Chapter 10: ArcSight Logger SmartMessage Pool (Encrypted) Destination	98
Configuring a Logger Pool Destination	98
Persisting SmartMessage Transport	101
Chapter 11: CEF Destinations	102
CEF File	102
File Rotation	103
Transformation Hub	103
CEF Syslog	106
Reconnect Feature for Load Balancing	108
CEF Encrypted Syslog (UDP)	109

Chapter 12: CSV File Destination	111
CSV File Installation	111
Event Data Rotation	113
Chapter 13: Raw Syslog Destination	114
Raw Syslog Overview	114
Appendix A: ArcSight Update Packs (AUPs)	115
ArcSight Content AUPs	115
ESM	115
ESM/Logger	115
Connector	116
Logger	116
Management Center	116
ESM Generated AUPs	116
User Categorization Updates	117
System Zones Updates	117
User Zones Updates	117
Appendix B: FIPS Compliant SmartConnectors	118
What is FIPS?	118
Which Connectors are Supported?	118
FIPS Compliant Connectors	118
FIPS Non-Compliant Connectors	119
Connectors Not Certified as FIPS Compliant	119
Connector Caveats	119
CEF Syslog as the Destination	119
Microsoft SQL JDBC Driver	120
Enable FIPS Support	120
Manually Enable FIPS Mode	120
Manually Enable FIPS Suite B Support	120
Password Management	121
Store Values	121
Entries for agent.properties File	121
Appendix C: Add Destinations	123
Client Authentication	123

Diagnosing Common Transformation Hub Issues	123
Transformation Hub Cluster Down	125
Pod Start Order	125
Cannot query ZooKeeper	125
Common Errors and Warnings in ZooKeeper logs	125
Common Errors and Warnings in Kafka logs	126
Configure an Transformation Hub Destination Without FIPS Mode	127
On the SmartConnector Server	127
Configure an Transformation Hub Destination with FIPS Mode	128
Step 1: On the Connector Server	129
Step 2: On the Transformation Hub Server	130
Step 3: On the Connector Server	130
Step 4: On the Transformation Hub Server	130
Step 5: On the Connector Server	130
Step 6: On the Transformation Hub Server	131
Configure an Transformation Hub Destination with Client Authentication in FIPS Mode	131
Step 1: On the Connector Server	132
Step 2: On the Transformation Hub Server	134
Step 3: On the Connector Server	135
Step 4: On the Transformation Hub Server	135
Step 5: On the Connector Server	135
Step 6: On the Transformation Hub Server	138
Configure an Transformation Hub Destination with Client Authentication in Non-FIPS Mode	138
Step 1: On the Connector Server	138
Step 2: On the Transformation Hub Server	140
Step 3: On the Connector Server	141
Step 4: On the Server	141
Step 5: On the Connector Server	142
Step 6: On the Transformation Hub Server	144
Configure an Transformation Hub Destination without Client Authentication in FIPS Mode	144
On the Transformation Hub Server	144
On the SmartConnector Server	145
Diagnostic Data and Tools	148
Installation Instructions for an Transformation Hub Destination	149
Master or Worker Nodes Down	149
Appendix C: Troubleshooting	151
Self-Healing for Unparsed Events	151

New Properties added	151
SmartConnector Commands Queue	152
Tuning Transformation Hub Performance	152
Increasing Stream Processor EPS	153
Increasing Kafka Retention Size/Time	153
Adding a New Worker Node	153
Verifying the Health of the Transformation Hub Cluster	153
What is FIPS?	154
Appendix D: Connector Frequently Asked Questions	155
Send Documentation Feedback	162

Chapter 1: About Connectors

This chapter provides an overview of ArcSight connectors and how they collect and send events (generated by various vendor devices) to the ArcSight ESM Manager, Logger, or other destinations.

A connector is an application that collects raw events from security devices, processes them into ArcSight security events, and transports them to destination devices. Connectors are the interface between the Manager and the network devices that generate ESM-relevant data on your network.

Connectors collect event data from network devices, then normalize it in two ways. First, they normalize values (such as severity, priority, and time zone) into a common format. Also, they normalize the data structure into a common schema. Connectors can filter and aggregate the events to reduce the volume sent to the Manager, ArcSight Logger, or other destinations, which increases ArcSight's efficiency and reduces event processing time.

Note: The device versions currently documented in individual SmartConnector configuration guides are versions that have been tested by ArcSight Quality Assurance. These are generally referred to as versions certified. For minor device versions that fall in between certified versions, it has been our experience that vendors typically do not make major changes to the event generation mechanism in minor versions; therefore, we consider these versions to be supported. Minor adjustments can be accommodated by parser overrides as needed. For example, Extreme Networks Dragon Export Tool versions 7.4 and 8.0 have been certified; Dragon Export Tool version 7.5 is considered to be supported.

In brief, connectors:

- Collect all the data you need from a source device, eliminating the need to return to the device during an investigation or audit.
- Parse individual events and normalize event values (such as severity, priority and time zone) into a common schema (format) for use by the ESM Manager.
- Filter out data you know is not needed for analysis, thus saving network bandwidth and storage space (optional).
- Aggregate events to reduce the quantity of events sent to the ESM Manager, increasing ArcSight's efficiency and reducing event processing time (optional).
- Categorize events using a common, human-readable format, saving you time and making it easier to use those event categories to build filters, rules, reports, and data monitors.
- Pass processed events to the ESM Manager.

Depending upon the network device, some connectors can issue commands to devices. These actions can be executed manually or through automated actions from rules and some data monitors.

Once connectors normalize and send events to the ESM Manager, the events are stored in the centralized ESM database. ESM then filters and cross-correlates these events with rules to generate meta-events. The meta-events then are automatically sent to administrators with corresponding Knowledge Base articles that contain information supporting their enterprise's policies and procedures.

Connectors process raw data generated by various vendor devices throughout an enterprise. Devices consist of routers, e-mail servers, anti-virus products, fire walls, intrusion detection systems (IDS), access control servers, VPN systems, anti-DoS appliances, operating system logs, and other sources that detect and report security or audit information.

Connectors collect a large amount of varying, heterogeneous information. Due to this variety of information, connectors format each event into a consistent, normalized ArcSight message, letting you find, sort, compare, and analyze all events using the same event fields.

Specific connector configuration guides document device-to-ESM event mapping information for individual vendor devices, as well as specific installation parameters and configuration information.

The following table lists destination settings that can be modified. These are functions that the connector performs on events. For details about how the following features work see [Configuring Destination Settings](#).

Feature	Description
Filtering and Data Reduction	Uses AND/OR based Boolean logic to determine what data is to be included from the device and what data is filtered out when the event is sent to the destination.
Aggregation	Compiles events with matching values into a single event, reducing the number of individual events the destination must evaluate.
Batching	Improves the destination performance by sending a collection of events at one time (rather than after each occurrence).
Time Error Correction	Synchronizes the time between the device and the connector, and between the connector and the destination.
Time Zone Correction	Corrects the local time zone, as necessary, to support device-time queries, correlation, and filters.
Categorizer	Assigns destination categories to an event.
Resolver	Attempts to resolve and reverse-resolve host names and addresses reported by a device.
Data Normalization	Converts each event produced by devices to a destination common event format message (or ArcSight message).
Logfu Command	A command that analyzes log files for troubleshooting problems by generating an HTML report (logfu.html) and including a graphical view of time-based log data. Logfu pinpoints the time of the problem and often the cause. If using PuTTY, you also need an X11 client on the machine from which you are connecting to the Linux machine.

Tip: You can deploy connectors on a device, on a separate host machine, or on the host machine

where the destination system resides.

Connectors both receive and retrieve information from network devices. If the device sends information, the connector becomes a receiver; if the device does not send information, the connector retrieves it.

Once an event is received by the connector, it adds device and event information to the event to complete the message, which is then sent to the configured destination.

Data Encryption

To follow new regulatory requirements that mandate that data leaving the connector machine to another destination be encrypted, you can use SecureData format preserving encryption.

When installing and configuring a connector, you can choose to enable this encryption. You will provide the URL of the encryption server, the identity and shared secret configured for SecureData, and the fields to be encrypted when configuring the connector. For optimum performance, the number of encrypted fields should be limited to 20 fields. If a proxy is enabled for the machine, a proxy host and port for http connection are also required.

See the ***SmartConnector Configuration Guide*** for the specific connector you are installing for a description of the format preserving parameters.

Notes:

- Once encryption is enabled, you cannot change any of the encryption parameters. To do so requires a new installation of the connector. If you install a connector without enabled encryption but want to do so later, you can Modify Connector Parameters through the wizard, enable encryption, and provide the encryption parameters.
- In deployments where multiple connectors are chained or cascaded before reaching the destination, the encryption should only be enabled at the very first connector.
- Encryption of address fields in the event is not supported. This includes IP addresses and MAC addresses.
- The input data must be at least three characters long to be encrypted if the data is all digits.
- This feature is supported only on Linux and Windows 64-bit platforms.
- Additional data fields cannot be selected for encryption.
- Although the connector and the destination can be set to FIPS-compliant mode for event data transfer between the connector and the destination configured, if encryption is enabled, the communication between the connector and secure server is not FIPS-compliant.
- Derived event fields cannot be chosen for encryption. If any of the derived fields need encryption, include the parent field for encryption.

Connector Data Collection

Connectors are specifically developed to work with network and security products using multiple techniques, including simple log forwarding and parsing, direct installation on native devices, SNMP, and syslog.

Data collection and event reporting formats for various connectors include:

- Log File Readers (including text and log file)
- Syslog
- SNMP
- Database
- XML
- Proprietary protocols, such as OPSEC

The ArcSight ESM Console, ESM Manager, and connectors communicate using HTTP (HyperText Transfer Protocol) over SSL (Secure Sockets Layer; also referred to as HTTPS).

Vendor device types for which connector are available include:

- Network and host-based IDS and IPS
- VPN, Firewall, router, and switch devices
- Vulnerability management and reporting systems
- Access and identity management
- Operating systems, Web servers, content delivery, log consolidators, and aggregators

Mapping to Vendor Events

Connectors collect the vendor-specific event fields logged by a network device. These fields are mapped to the ArcSight data fields within the connector, based on the ArcSight ESM schema prior to being forwarded to the configured destination.

For specific mappings between the connector data fields and supported vendor-specific event definitions, see the configuration guide for the device-specific connector. For example, for mappings for the SmartConnector for Cisco PIX/ASA Syslog, see the configuration guide for the SmartConnector for Cisco PIX/ASA Syslog.

General mappings for ArcSight Common Event Format connectors are documented in the **ArcSight Common Event Format (CEF) Guide**, also known as "Implementing ArcSight Common Event Format (CEF)". To access this document, go to <https://community.softwaregrp.com/t5/ArcSight-Connectors/tkb-p/connector-documentation>.

For mappings for a certified CEF vendor's connector, see their product documentation, available from the Micro Focus Enterprise Security Technology Alliances site on Protect 724 at <https://community.softwaregrp.com/t5/Security-Technology-Alliances/ct-p/technology-alliances>.

Filter and Aggregate Events

Filter conditions to focus the events passed to the destination according to specific criteria can be added during SmartConnector installation and configuration. For example, you can use filters to sort out events with certain characteristics, from specific network devices, or generated by vulnerability scanners. Events that do not meet the connector filtering criteria are not forwarded.

The connector can be configured to aggregate (summarize and merge) events that have the same values in a specified set of fields, either for a specified number of times or within a specified time limit.

Connector aggregation compiles events with matching values into a single event. The aggregated event contains only the values the events have in common plus the earliest start time and latest end time. This reduces the number of individual events that must be evaluated.

For example, suppose the connector is configured to aggregate events with a certain Source IP and Port, Destination IP and Port, and Device Action whenever the events occur 10 times in 30 seconds. If ten events with these matching values are received by the connector within that time frame, they are grouped together into a single event with an aggregated event count of 10.

If the 30-second time frame expires and the connector has received only two matching events, the connector creates a single aggregated event with an aggregated event count of two. If 900 matching events were to come in during the 30 seconds, the connector would create 90 aggregated events, each with an aggregated event count of 10.

Firewalls are a good candidate for aggregation because of the volume of events with similar data coming in from multiple devices.

Note: When selecting **Aggregate Events**, the **Preserve Raw Event** feature is disabled.

For filtering events prior to ingestion refer to [Customized Events Filtering](#).

Connector Types

Connectors are the interface between the ESM Manager and the network devices that generate ESM-relevant data on your network.

Connectors are generally one of the following types:

- API Connectors
- Database Connectors

- FlexConnectors
- File Connectors
- Microsoft Windows Event Log Connectors
- Model Import Connectors
- Scanner Connectors
- SNMP Connectors
- Syslog Connectors

Connectors collect event data from network devices, then normalize this data. These connectors normalize values (such as severity, priority, and time zone) into a common format. The data structure is normalized into a common schema. Connectors can filter and aggregate events to reduce the volume sent to the destination, which increases efficiency and reduces event processing time.

For installation information and device-specific configuration and mapping information, see the connector Configuration Guide for the specific device.

File Connectors

There are two primary types of log file connector, **Real Time** and **Folder Follower**:

Real Time

These connectors can continue to follow a log file that retains its name or changes its name based upon the current date and other factors. The type of real time file connector is based upon the number of files monitored by the connector. There are connectors that monitor a single log file and connectors that monitor multiple log files.

Real Time log file connectors can read normal log files in which lines are separated by a new line character as well as fixed length records in which a file consists of only one line but multiple records of fixed length.

Folder Follower

Folder follower connectors can monitor files copied to a folder. There are connectors that monitor a single log file in a folder and connectors that monitor log files recursively.

.txt and **.xml** file types are supported by connectors; which type depends upon the particular device. Most of the scanner file connectors, such as Nessus, and NeXpose, are in XML format.

The type of log file connector is not usually part of the connector name unless both types of connector exist for a particular device.

File connectors are normally installed on the device machine, but when the monitored files are accessible through network shares or NFS mounts, the connectors can be installed on remote machines.

For some connectors, a trigger file is required to tell the connector when the file is complete and ready for processing. Typically, this is the same file name with a different extension. Files are renamed by default to increments such as **.processed**, **.processed.1**, and so on.

Generally, the only parameter required at installation is the location of the log file or files (the absolute path). When default file paths are known, they are displayed in the installation wizard.

Note: To rename or delete log files, file folders require permissions for the connector.

Database Connectors

Database connectors use SQL queries to periodically poll for events. Connectors support major database types, including MS SQL, MS Access, MySQL, Oracle, DB2, Postgres, and Sybase.

During installation, the installation wizard asks, at a minimum, the following parameter values:

- JDBC Driver
- JDBC Database URL
- Database User
- Database Password

The database user must have adequate permission to access and read the database. For Audit database connectors, such as SQL Server Audit DB and Oracle Audit DB, system administrator permission is required.

In addition to connectors supporting event collection from a single database, some database connectors support multiple database events such as the Microsoft SQL Server Multiple Instance DB connector. Others collect events from scanner databases, such as the connector for McAfee Vulnerability Manager DB.

There are three major types of database connector:

Time-Based

Queries use a time field to retrieve events found since the most recent query time until the current time.

ID-Based

Queries use a numerically increasing ID field to retrieve events from the last checked ID until the maximum ID.

Job ID-Based

Queries use Job IDs that are not required to increase numerically. Processed Job IDs are filed in such a way that only new Job IDs are added. Unlike the other two types of database connector, Job IDs can run in either Interactive mode or Automatic mode.

API Connectors

API connectors use a standard or proprietary API to pull events from devices. In most cases, a certificate must be imported from the device to authenticate connector access to the device. There are also a number of configuration steps required on the device side.

During installation, the following types of parameters are required, although each device's parameters are specific to its API:

- Device IP
- Service Port
- Event types to be pulled
- Certificate information
- Information specific to the particular API

SNMP Connectors

SNMP Traps contain variable bindings, each of which holds a different piece of information for the event. They are usually sent over UDP to port 162, although the port can be changed.

SNMP connectors listen on port 162 (or any other configured port) and process the received traps. They can process traps only from one device with a unique Enterprise OID, but can receive multiple trap types from this device.

SNMP is based upon UDP, so there is a slight chance of events being lost over the network.

Although there are still some SNMP connectors for individual connectors, most SNMP support is provided by the SmartConnector for SNMP Unified. Parsers use the knowledge of the MIB to map the event fields, but, unlike some other SNMP-based applications, the connector itself does not require the MIB to be loaded.

Microsoft Windows Event Log Connectors

System administrators use the Windows Event Log for troubleshooting errors. Each entry in the event log can have a severity of **Error**, **Warning**, **Information**, plus **Success Audit** or **Failure Audit**.

There are three default Windows Event Logs:

- Application log (tracks events that occur in a registered application)
- Security log (tracks security changes and possible breaches in security)
- System log (tracks system events)

There are three connectors for Microsoft Windows Event Log:

- **SmartConnector for Microsoft Windows Event Log, SmartConnector for Microsoft Windows Event Log – Native** and **SmartConnector for Microsoft Windows Event Log – Unified**: these connectors can connect to local or remote machines (Windows), inside a single domain or from multiple domains, to retrieve and process security and system events.

For details about the Unified connector, see the configuration guide for the SmartConnector for Microsoft Windows Event Log – Unified. For mappings, see the document *SmartConnector for Microsoft Windows Event Log – Unified Windows 2008/2012 Security Event Mappings*. For details about the Native

connector, see the configuration guide for the SmartConnector for Microsoft Windows Event Log -- Native. For mappings, see the document *SmartConnector for Microsoft Windows Event Log - Native Windows Security Event Mappings*.

These connectors provide support for partial event parsing based upon the Windows event header for all System and Application events. Support for a FlexConnector-like framework that lets users create and deploy their own parsers for parsing the event description for all System and Application events is also provided.

Some individual Windows Event Log applications are supported by the connectors for Microsoft Windows Event Log, Microsoft Windows Event Log – Unified and Microsoft Windows Event Log – Native connectors, for which Windows Event Log application or system support has been developed. See the configuration guides for these connector for a list of application and system events supported.

Syslog Connectors

Syslog messages are free-form log messages prefixed with a syslog header consisting of a numerical code (facility + severity), timestamp, and host name. They can be installed as a syslog daemon, pipe, or file connector. Unlike other file connectors, a syslog connector can receive and process events from multiple devices. There is a unique regular expression that identifies the device.

- **Syslog Daemon** connectors listen for syslog messages on a configurable port, using port 514 as a default. The default protocol is UDP, but other protocols such as Raw TCP are also supported. It is the only syslog option supported for Windows platforms.
- **Syslog Pipe** connectors require syslog configuration to send messages with a certain syslog facility and severity.

The Solaris platform tends to under perform when using Syslog Pipe connectors. The operating system requires that the connector (reader) open the connection to the pipe file before the syslog daemon (writer) writes the messages to it. When using Solaris and running the connector as a non-root user, using a Syslog Pipe connector is not recommended. It does not include permissions to send an HUP signal to the syslog daemon.

- **Syslog File** connectors require syslog configuration to send messages with a certain syslog facility and severity. For high throughput connectors, Syslog File connectors perform better than Syslog Pipe connectors because of operating system buffer limitations on pipe transmissions.
- **Raw Syslog** connectors generally do no parsing and takes the syslog string and puts it in the **rawEvent** field as-is. The Raw Syslog destination type takes the **rawEvent** field and sends it as-is using whichever protocol is chosen (UDP, Raw TCP, or TLS). The Raw Syslog connector is always used with the Raw Syslog destination. The event flow is streamlined to eliminate components that do not add value (for example, with the Raw Syslog transport the category fields in the event are ignored, so the categorization components are skipped). If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the syslog data (for source and timestamp).
- **Syslog NG Daemon** connectors support Syslog NG version 3.0 for BSD syslog format. Support is

provided for collection of IETF standard events. This connector is capable of receiving events over a secure (encrypted) TLS channel from another connector (whose destination is configured as CEF Syslog over TLS), and can also receive events from devices.

- **CEF Encrypted Syslog (UDP)** connectors allow connector-to-connector communication through an encrypted channel by decrypting events previously encrypted through the CEF Encrypted Syslog (UDP) destination. The CEF connector lets ESM connect to, aggregate, filter, correlate, and analyze events from applications and devices that deliver their logs in the CEF standard, using the syslog transport protocol.

UNIX supports all types of syslog connector. If a syslog process is already running, you can end the process or run the connector on a different port.

Because UDP is not a reliable protocol, there is a slight chance of missing syslog messages over the network. Generally, TCP is a supported protocol for syslog connectors.

There is a basic syslog connector, the connector for UNIX OS Syslog, which provides the base parser for all syslog sub-connectors.

For syslog connector deployment information, see the connector Configuration Guide for UNIX OS Syslog. For device-specific configuration information and field mappings, see the connector configuration guide for the specific device. Each syslog sub-connector has its own configuration guide.

During connector installation, for all syslog connectors, choose **Syslog Daemon**, **Syslog Pipe**, or **Syslog File**. The names of the syslog sub-connectors are not listed.

Scanner Connectors

There are two types of scanner connector, those whose results are retained within a file, and those retrieved from a database. Results for XML scanner connectors are retained in a file, making them log file connectors.

Other scanners deposit their scanned events in a database and are treated as database connectors, requiring the same installation parameters as database connectors.

Scan reports are converted into base events, which, for ESM destinations, can be viewed on the Console, and aggregated meta events, which are not shown on the Console. Meta events create assets, asset categories, open ports, and vulnerabilities on the Console.

Scanner connectors can run in either of two modes, automatic or interactive.

Interactive mode

In Interactive mode, a graphical user interface shows the reports or log files available for import from the configured log directory. Choose reports to send to the connector by checking the box for **Send** for individual log files and clicking **Send to ArcSight**.

Automatic mode

Automatic mode is designed to be used in conjunction with an automated procedure to periodically run scans. The procedure, or shell script, should execute the scanner periodically and save a report in **.cef** format. At the end of the scan, after the report is saved, an empty file called **<reportname>.cef_ready** should be created, which indicates to the connector that the .cef report is ready for importing. The connector continues to search for **.cef_ready** files and process the corresponding .cef reports. The processed reports are renamed to **<original report file>.cef_processed**.

Other than the operating mode, other parameter values required for scanner installation depends upon whether a file or database connector has been implemented. For file connectors, the absolute path to and name of the log file is required. For database connectors, see ["Database Connectors" on page 17](#).

FlexConnectors

FlexConnectors let you create custom connectors that can read and parse information from third-party devices and map that information to the ArcSight event schema. When creating a custom connector, you define a set of properties (a configuration file) that identify the format of the log file or other source that is imported into the ESM Manager or Logger.

The FlexConnector framework is a software development kit (SDK) that lets you create a connector tailored to the devices on your network and their specific event data. For complete information about FlexConnectors and how to use them, see the ***FlexConnector Developer's Guide***.

Model Import Connectors

Rather than collecting and forwarding events from devices, Model Import Connectors import user data from an Identity Management system into ArcSight ESM. See individual configuration guides for Model Import Connectors on Protect724 for information about how these connectors are used.

Model Import Connectors extract the user identity information from the database and populate the following lists in ESM with the data:

- Identity Roles Session List
- Identity Information Session List
- Account-to-Identity Map Active List

These lists are populated dynamically, which means that, as the identity data changes in the Identity Manager, the data in the lists is updated when you refresh the session list.

Other Connectors

Connectors that Use Multiple Mechanisms

Some connectors use multiple mechanisms. For example, the connector for Oracle Audit Database monitors both the database tables and audit files.

Connectors that Use TCP in Special Formats

Examples of connectors use TCP in special formats:

IP NetFlow (NetFlow/J-Flow)

Retrieves data over TCP in a Cisco-defined binary format.

ArcSight Streaming Connector

Retrieves data over TCP from Logger in an ArcSight-proprietary format.

Management Center

ArcSight Management Center manages and monitors a range of ArcSight products, such as Loggers, and other ArcSight Management Centers. In this guide, these products are referred to as Management Center.

The Management Center centralizes connector management and offers unified control of connectors on local and remote Management Centers as well as software-based connectors installed on remote hosts.

Management Center includes on-board connectors that connect event sources to destinations such as Logger and ESM.

The Management Center delivers the following features and benefits:

- Supports bulk operations across all connectors and is particularly desirable in ESM deployments with a large number of connectors, such as a Managed Security Services Provider (MSSP).
- Provides an ESM-like connector management facility in Logger-only environments.
- Provides a single interface through which to configure, monitor, tune, and update connectors. The Management Center does not receive events from the connector it manages, and this allows for management of many connectors at one time. The Management Center does not affect working connectors unless it is used to change their configuration. In some cases, the connector is commanded to restart.

For a complete list of all connectors supported by the Transformation Hub, see its Release Notes. You can also visit the Community site at <https://community.softwaregrp.com/t5/ArcSight/ct-p/arcSight>. ArcSight adds new connectors regularly.

See ["Connectors with Management Center" on page 68](#) for further details.

ArcSight Logger

Logger is an event data storage appliance optimized for extremely high event throughput. Logger stores security events onboard in compressed form, but can always retrieve unmodified events on demand for forensics-quality litigation data.

Logger can be deployed stand-alone to receive events from syslog messages or log files, or to receive events in Common Event Format from connectors. Logger can forward events to ESM. Multiple Loggers work together to support high sustained input rates. Event queries are distributed across a peer network of Loggers. See ["ArcSight Logger SmartMessage \(encrypted\) Destination" on page 91](#) for details on the relationship between connectors and Logger.

Instant Connector Deployment

The Instant Connector Deployment feature allows a remote silent installation of connectors from the ArcSight Management Center (ArcMC) Deployment View on a host and does not require a connector to have been previously installed.

The main goal of Instant Connector Deployment is to make installation of connectors a simpler process for enterprise customers who deploy to a high number of servers and may install multiple connectors per server. All of the installation information is captured up front, then deployed and installed to many target nodes through ArcMC.

For more information, see the *ArcSight Management Center Administrator's Guide*.

Connectors in Transformation Hub

Connectors in Transformation Hub supports ArcSight customers who want to have large-scale distributed ingestion pipelines with 100% availability, where data from any existing or new source at any scale can be ingested while maintaining enterprise level robustness.

Transformation Hub can take messages with raw data collected from any source the ArcSight connector framework understands and automatically perform the data ingestion processing currently done by connectors, but deployed and managed at scale as Transformation Hub processing engines.

Users deploy the Transformation Hub using the ArcSight Installer and Management Center to achieve the desired layout. New topics can be created in Management Center and designated to process raw data from a particular technology framework with output into a specific format.

The connector technology in Transformation Hub performs all processing a connector would normally do: parser selection, normalization, main flow, destination specific flows, and categorization, as well as applying network zoning and Agent Name resolution.

For more information, see the *ArcSight Transformation Hub Administrator's Guide* and the *ArcSight Management Center Administrator's Guide*.

Chapter 2: Planning for Deployment

Deployment of a connector is based upon the requirements of your network security enterprise. This section outlines possible ArcSight deployments based upon different scenarios.

The scenarios and deployments shown here are only examples of how you might introduce ESM into your enterprise. ESM is not limited to just these scenarios and deployments.

Deployment Overview

ArcSight components install consistently across UNIX, Windows, and Macintosh platforms. Whether a host is dedicated to the ArcSight Database, Manager, Console, or other component, ESM software is installed in a directory tree under a single root directory on each host (DBMS and other third-party software is not necessarily installed under this directory, however.) The path to this root directory is referred to as **\$ARCSIGHT_HOME**.

In connector documentation, the 'current' directory is specified rather than presumed to be part of the **\$ARCSIGHT_HOME** location, and the path separator is a backslash (\) (for example, **\$ARCSIGHT_HOME\current**). This is consistent with connector configuration guide information, and also underscores the fact that connectors are not installed on the same machine as the remaining ESM components. Rather, they are typically installed on the same machine as the device whose activity will be monitored.

The directory structure below **\$ARCSIGHT_HOME** is standardized across components and platforms. ArcSight software is generally available in the **\$ARCSIGHT_HOME\current\bin** directory. Properties files, which control the ArcSight configuration, are found in **\$ARCSIGHT_HOME\config** and log files are written to **\$ARCSIGHT_HOME\logs**.

Connectors collect and process the data generated by various vendor devices throughout your enterprise. Devices consist of routers, e-mail logs, anti-virus products, fire walls, intrusion prevention systems (IPS), access control servers, VPN systems, antiDoS appliances, operating system logs, and other sources where information about security threats are detected and reported.

Connectors collect a vast amount of varying, heterogeneous information. When a connector receives an event, it completes the message by adding device information, then forwarding the event to various ArcSight components.

Supported Platforms

For information about supported platforms, see the ArcSight SmartConnector Platform Support document that is shipped with each connector release. Only differences to the support detailed in that document are specified in the device's connector configuration guide.

Deployment Scenarios

You can install connectors on the ESM Manager machine, the machine hosting ArcSight Management Center, a host machine, or a device. Based upon configuration, connectors also can receive events over the network using SNMP, HTTP, syslog, proprietary protocols (such as OPSEC), or direct database connections to the device's repository (such as ODBC or proprietary database connections).

The best deployment scenario for your system depends upon the connector type, your network architecture, and your operating system.

- Scenarios for syslog deployment are documented in the ***Connector for UNIX OS Syslog Configuration Guide***.
- Scenarios for deploying Windows Event Log connectors are documented in the configuration guides for the SmartConnector for Microsoft Windows Event Log Unified and Native and SmartConnector for Windows Event Log.

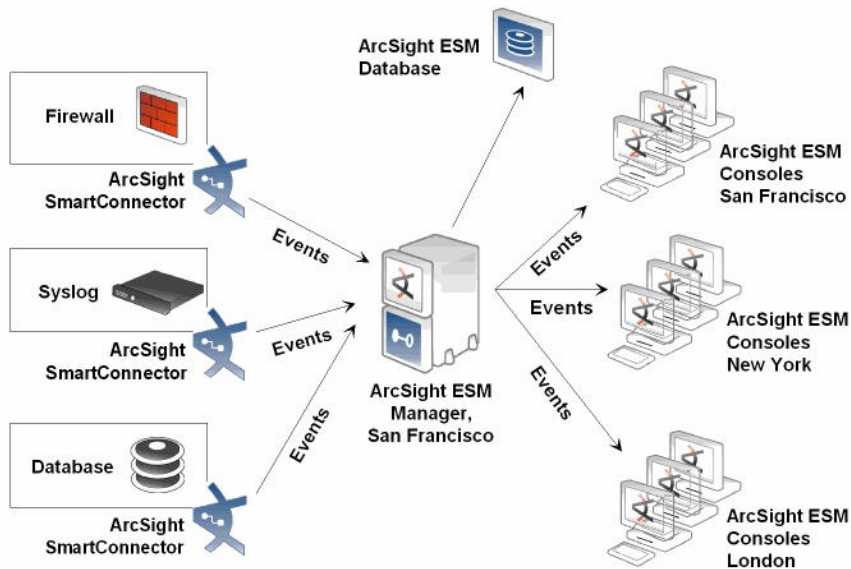
Deployment Scenario One

In this scenario, there are three connectors residing on three different devices: a firewall, an IPS, and a UNIX operating system. These connectors receive information from the devices or their logs and send captured events to the Manager based upon the connector configuration.

Once events are received by the Manager, it cross-correlates the events using rules, and sends meta-events to the Database and to any Consoles that access the database.

The ESM Manager also can perform preset actions. Events and meta-events within the Database can be played back using the ***Replay*** channel to investigate, analyze, or create a report about event history.

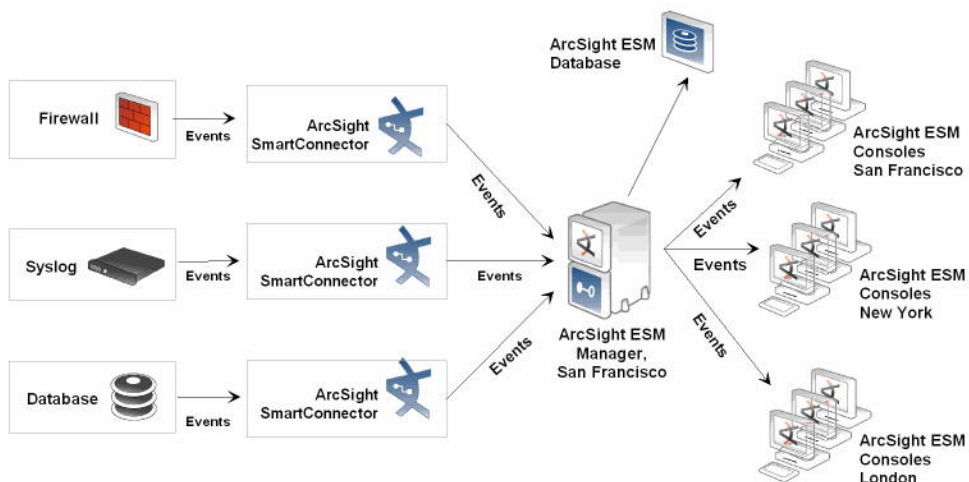
Three Connectors Residing on Three Devices



Deployment Scenario Two

This scenario is the same as the first, except that the three connectors reside on a host machine rather than the device itself. The connector need not reside on the device in order to retrieve information from that device. The connector functions as before, and the ArcSight ESM Manager and Database perform the same functions.

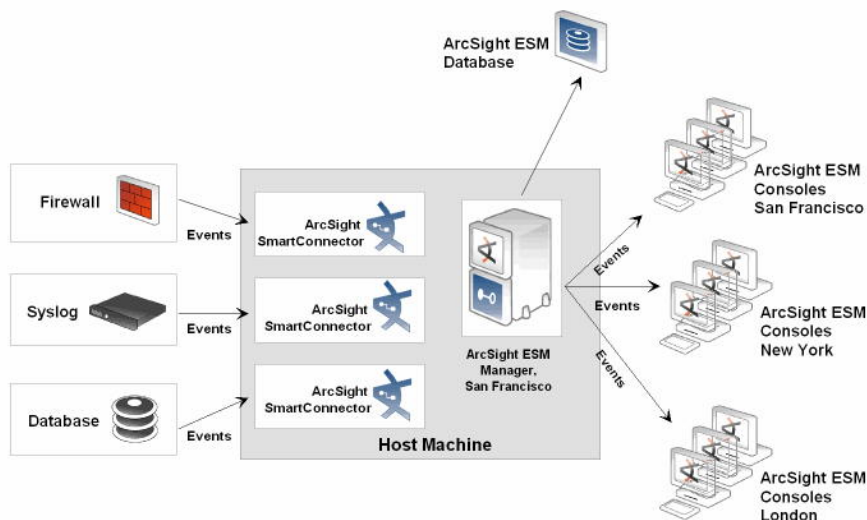
Three Connectors Residing on a Host Machine



Deployment Scenario Three

In this scenario, the connectors reside on the ESM Manager itself, not on a host machine, but still retrieve events from devices in the network. The processing performed by the ArcSight connector, Manager, and Consoles are identical to the other scenarios.

Three Connectors Residing on an ESM Manager



Deployment Scenario Four

In this scenario, any of the previous scenarios are implemented, and the connectors are configured to send events to Logger. From Logger, events can be forwarded on to ESM.

Estimating Storage Requirements

Understanding the range of devices and connectors you want to deploy helps in estimating your daily event volume. Log file size is not accurate enough; you need to know how many events are generated during an average day. This varies by the type of device. Not only do different devices generate different event volumes, they also respond differently to various event aggregation policies.

The average size of the data stored for each event depends upon the **turbo mode** (**Fastest**, **Faster**, or **Complete**) specified for a particular connector. For detailed information on turbo modes, see ["Understanding ArcSight Turbo Modes" on the next page](#).

Connectors can aggregate events to reduce event traffic. An event that repeats every 500 ms, for example, can be represented by a single event that fires every ten seconds, producing a 20:1 event compression.

Individual connectors can be configured to aggregate events in this manner, reducing event traffic to the ESM Manager and the storage requirements in the ESM Database.

In a distributed environment with multiple ESM Managers, the event volume metric must consider both the connector feeds to the ESM Manager and the event forwarding from other ESM Managers.

Understanding ArcSight Turbo Modes

You can accelerate the transfer of sensor information through connectors by choosing one of three turbo modes (**Fastest**, **Faster**, or **Complete**).

The **Fastest** mode requires the fewest bytes and is most suited to devices such as firewalls, which have relatively little event data. The **Faster** mode is the ESM Manager default, and requires less storage space. Rich event data sources, such as a network operating system, might use **Complete** mode, the connector default. The **Complete** mode passes all the data arriving from the device, including any custom or vendor-specific (for example, “additional”) data.

You can configure connectors to send more or less event data on a per-connector basis, and the ESM Manager can be set to read and maintain more or less event data, independent of the connector setting.

Some events require more data than others. For example, operating system logs often capture a considerable amount of environmental data that may not be relevant to a particular security event. Firewalls, on the other hand, typically report only basic information.

ArcSight defines turbo modes as follows:

Mode	Description
Fastest (Mode 1)	Recommended for simpler devices, such as firewalls.
Faster (Mode 2)	ESM Manager default. Eliminates all but a core set of event attributes to achieve the best throughput. Because the event data is smaller, it requires less storage space and provides the best performance.
Complete (Mode 3)	Connector default. All event data arriving at the connector, including additional data, is maintained.

When a turbo mode is not specified, Mode 3, Complete, is the default. Versions of ESM prior to version 3.0 run in turbo mode **Complete**.

The ESM Manager uses its own turbo mode setting when processing event data. If a connector is set at a higher turbo mode than the ESM Manager, it reports more event data than the ESM Manager requires. The ESM Manager ignores these extra fields.

However, if an ESM Manager is set at a higher turbo mode than the connector, the connector has less event data to report to the ESM Manager. The ESM Manager maintains fields that remain empty of event data.

Both situations are normal in real-world scenarios because the ESM Manager configuration must reflect the requirements of a diverse set of connectors.

Chapter 3: Installing Connectors

When you are ready to install a connector, see the individual connector's configuration guide for information specific to the device the connector is monitoring. For example, when installing a connector for Microsoft Windows Event Log, see the connector Configuration Guide for Microsoft Windows Event Log, Unified, or Native.

Individual configuration guides provide information about how to configure the particular device to enable connector event collection, installation instructions as well as parameters required for installation, and customized mappings of vendor device event fields to ArcSight fields.

Note: If you are using the Linux Red Hat 6.x or later platforms, ensure that you have these libraries or packages installed before installing a connector:

- X libraries
- glibc
- libXext
- libXrender
- libXtst
- unzip
- fontconfig \ dejavu-sans-fonts

When installing the 32-bit SmartConnector executable on 64-bit machines, the 32-bit versions of glibc, libXext, libXrender, and libXtst must be installed as well as the 64-bit versions.

Installing the Connector from an Executable

When you perform an installation, you are asked to specify the connector you want to install. Download the executable and the zip file of connector documentation. Each connector has a separate configuration guide that provides specific instructions on installing the connector and configuring any associated devices, as well as device-to-ESM event mapping information for individual vendor devices, specific installation parameters, and device configuration information.

Note: On Windows, do not install in a directory with an open or close parenthesis () character in the name.

Also, see the ArcSight SmartConnector Release Notes, which describe new product features, latest updates, and known product issues and workarounds. For information regarding operating systems and platforms supported, see the SmartConnector Platform Support document.

Note: The 64-bit installation executables contain a subset of available SmartConnector. See the 64-bit SmartConnector installer for your platform for the list of available connectors, or see the document “SmartConnector 64-Bit Support” document available on Protect 724 or in the SmartConnector Configuration Guide zip file available for download on the Micro Focus SSO Site.

If you have been running a 32-bit SmartConnector, you cannot upgrade to the 64-bit version. To run the 64-bit SmartConnector implementation, perform a new installation.

You will be prompted for an installation destination. By default, the destination is ArcSight Manager (encrypted). For details on destinations, see ["Connector Destinations Overview" on page 72](#). If you need information on a FIPS-compliant solution, see the specific connector configuration guide. The Parameters window requests specific parameters for the particular connector you selected. These parameters vary depending upon the device and are described and explained in the connector configuration guide for the selected connector.

It is a good practice to develop and use a standard naming convention to specify directory locations, file names, and menu option names for the connectors you install. Typically, if you install multiple connectors on a particular machine, you should install each connector in a separate directory.

Connectors can also be installed from the command line (see ["Installing Connectors from the Command Line" below](#)) or using silent mode, which answers the wizard questions from a properties file (see ["Installing Connectors in Silent Mode" below](#)).

Installing Connectors from the Command Line

To install connectors without using the graphical user interface wizard, enter **-i console** on the command line when you invoke the self-extracting archive. Follow the instructions in the command window.

When the installation has successfully completed, manually run the configuration program by executing **runagentsetup**.

Installing Connectors in Silent Mode

You can run the connector installation in silent mode, in which answers to wizard questions are provided by a Properties file. This feature is useful for deploying a large number of identical connectors.

To use this feature, first install and configure one connector using the graphical-user interface or the command line. While configuring the first connector, record its configuration parameters in a Properties file. To install all other connectors in silent mode, use the Properties file you created to provide configuration information.

Tip: ArcSight recommends creating and testing the Properties file on a system other than your in-service, production environment.

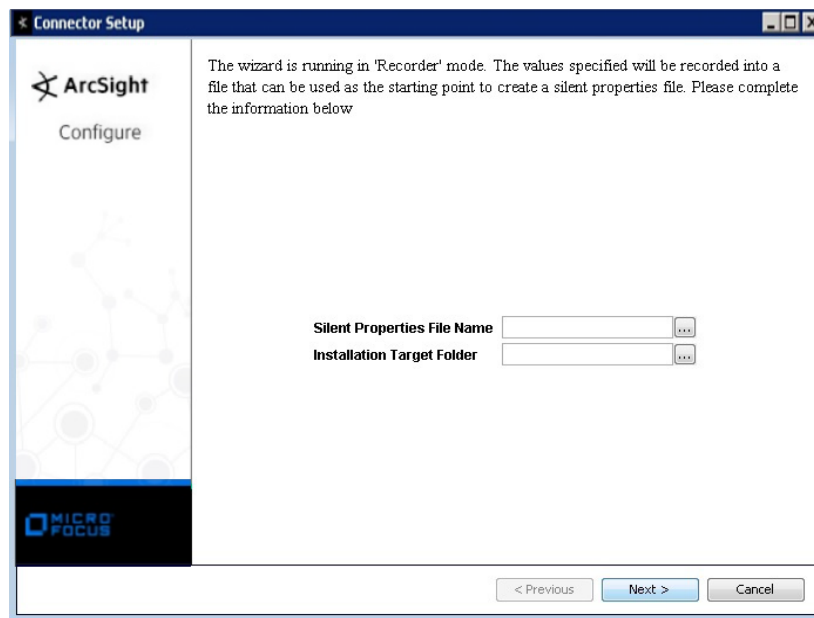
To record the configuration of a SmartConnector to a Properties file:

1. Run the connector Configuration Wizard to extract and install the connector core files. When the wizard asks you to choose **Add a Connector** or **Set Global Parameters**, click **Cancel**.
2. From a command prompt window (from the **ARCSIGHT_HOME\current\bin** directory), enter the following command to launch the connector Configuration Wizard in record mode:

On Unix and Linux: `./runagentsetup.sh -i recorderui`

On Windows: `runagentsetup.bat -i recorderui`

3. On the window displayed, enter the **Silent Properties File Name**. Enter the name of the **Installation Target Folder** to select a location.



4. Continue through all connector Configuration Wizard windows. The wizard creates a Properties file using the name and location you specified.

Note: The properties file that you create will show passwords in readable text.

5. Select **Exit** and click **Next** at the end of the setup process to ensure that the properties file is created.

Perform the remaining steps on the system on which you want to install the SmartConnector in silent mode:

1. Ensure that the configuration on the system on which you want to install the connector in silent mode matches that of the machine on which you created the properties file. Otherwise, the installation will fail.
2. Copy the Properties file from the other system to your current system, preferably to the same directory where you downloaded the installation file.
3. Open the Properties file in an editor of your choice.
4. Find the **USER_INSTALL_DIR** property in the file and make sure that the path value is the **absolute**

path to the location where you want to install the connector on this system.

USER_INSTALL_DIR=C:\Program Files\ArcSightSmartConnectors

Note: The colon (:) and backslash (\) characters must be preceded by a backslash (\).

5. Find the **ARCSIGHT_AGENTSETUP_PROPERTIES** property in the file and make sure that the path value is the **absolute** path to the location where you copied the Properties file on this system.

For example, if you copied the Properties file to **C:\properties_files\silent.properties**, the path value should be as follows:

ARCSIGHT_AGENTSETUP_PROPERTIES=C:\\properties_files\\silent.properties

6. Modify the properties as needed. For example, modify the **connectordetails.name** property in the file and change its value to the name of the connector you are going to install in silent mode. The following is an example of a properties file:

```
#=====
# Panel 'connectordetails'
#=====
# Enter the connector details.
#
# Name
connectordetails.name=The Name
# Location
connectordetails.location=The Location
# DeviceLocation
connectordetails.devicelocation=The Device Location
# Comment
connectordetails.comment=The Comment
#=====
```

You can edit any property (Manager Information, user credentials) in the Properties file to suit your needs.

Definitions of properties:

- **connectordetails.name:** The name of the connector in ESM.
- **connectordetails.location:** The name of the folder that contains the connector in ESM.
- **connectordetails.devicelocation:** The location of the machine on which ESM is installed.
- **connectordetails.comment:** Comments that were added about the connector.

7. Save the Properties file.
8. Download the connector installation file appropriate for your platform.
9. Run the following command to install the new connector in silent mode:

```
ArcSight_Agent_install_file -i silent -f <path_to properties_
file>\properties_filename
```

The command launches the InstallShield program and installs the connector silently.

Example: To install a connector on Windows platform with the property file name **silent_properties**, enter:

ArcSight-3.5.x.nnnn.y-Agent-Win.exe -i silent -f silent_properties

Note: After running the silent install, the original command in the **runagentsetup.bat** file is modified after specifying the Silent Install answer file.

To correct the problem, manually edit and remove the entries between the double quotes and return to the default setting. There should be no entries between the second double quotes. For example, the modified script may look like this:

```
call arcsight.bat agentsetup -c -i "SILENT" -f "C:\ArcSight\silent_properties_AD" %*
```

After manually editing the entries, it should look like this:

```
call arcsight.bat agentsetup -c -i "SWING" -f "" %*
```

To avoid this issue:

Extract first and use the **silent_properties** file to configure. Run the command similar to following:

```
<connector_installpath>\current\bin\arcsight.bat agentsetup -c -i silent -f 2_addwinc
```

Then, the **runagentsetup.bat** file would not contain the **silent_properties** and the path will be correct.

Caution: It is important to know:

- After installing connector, configure your system's default file permissions so that files created by ArcSight (events, log files, and so on) are reasonably secure.
- On UNIX systems, file permissions typically are set by adding the **umask** command to your shell profile. An umask setting of 077, for example, would deny read or write file access to any but the current user. An umask setting of 000 creates an unnecessary security hole.

Uninstalling a Connector

Before uninstalling a connector that is running as a service or daemon, first stop the service or daemon. Also, be sure to remove the service files using `$ARCSIGHT_HOME/current/bin/arcsight agentsvc -r` before uninstalling the connector.

The Uninstaller does not remove all the files and directories under the connector home folder. After completing the uninstall procedure, manually delete these folders.

To uninstall on Windows:

1. Open the Start menu.
2. Run the Uninstall SmartConnectors program found under All Programs -> ArcSight SmartConnectors (or the name you used for the folder during connector installation).
3. If connectors were not installed on the Start menu, locate the `$ARCSIGHT_HOME/current/UninstallerData` folder and run: **Uninstall_ArcSightAgents.exe**

Note: To perform a silent uninstall, run the command with the following parameters:
Uninstall_ArcSightAgents.exe -i silent

To uninstall on UNIX hosts:

1. Change to the `$ARCSIGHT_HOME/UninstallerData` directory.
2. Run the command: `./Uninstall_ArcSightAgents`.

Note: The UninstallerData directory contains a file `.com.zerog.registry.xml` with Read, Write, and Execute permissions for everyone. On Windows platforms, these permissions are required for the uninstaller to work. However, on UNIX platforms, you can change the permissions to Read and Write for everyone (that is, 666).

Note: To perform a silent uninstall, run the command with the following parameters:
`./Uninstall_ArcSightAgents -i silent`

Upgrading Connectors from ESM

Connectors occasionally may require upgrade. This process can be performed locally or remotely, although remote upgrades from the Console are supported only on Windows, Linux, and Solaris platforms.

Note: For connectors running on windows platforms, there is a known limitation for upgrading the connector from its ESM destination.

As part of the connector upgrade, some folders or files are moved from the old to the new version. Because Microsoft Windows locks the folders or files even they are opened for a read, upgrades could fail if locked folders or files associated with the connector installation are accessed during the upgrade. To prevent this issue, start the connector from **Start > Programs**, so that no windows are opened to run the connector, thus reducing the possibility of locked folders or files.

To upgrade:

1. From the Micro Focus SSO site, download the latest connector upgrades to the Manager. Upgrade version files are delivered as **.aup** files (a compressed file set).
2. Copy the **.aup** file to **ARCSIGHT_HOME\updates** on a running Manager. The Manager automatically unzips the **.aup** file and copies its content to **ARCSIGHT_HOME\repository**.
3. From the Console, select connectors to be upgraded (one at a time) and launch the **upgrade** command for each of them.

Caution: It is important to know:

- If you have installed multiple connectors in a single JVM, select the first connector installed in the JVM (if you select any other connector the upgrade fails) and launch the **upgrade** command; this action upgrades all connectors in the JVM.
- If your connector has multiple Manager destinations, you must perform this process from the primary Console. Any attempt to upgrade from a secondary or non-primary Console destination will fail.

4. Upon receipt of the **upgrade** command, the selected connectors upgrade themselves, restart, and send upgrade results (success or failure) back to the Console through the Manager.

Upgrade Notes

- If the upgrade is successful, the new connector starts and reports successful upgrade status.
- If the upgraded connector fails to start, the original connector restarts automatically as a failover measure.

Tip: You may want to know:

- Should this happen, you can review the related logs. Choose **Send Command -> Tech Support -> Get Upgrade Logs** from the Console menus.
- You can also use the Send Logs Wizard to collect and send logs, including upgrade logs, to support for help.

- Connectors automatically determine their upgrade status when they start.
- When upgrading connectors, be sure to download current versions of the connector Configuration

Guides from the support website. These are the most current configuration guides available and contain information specific to the connector device.

- Administrative permission is required to upgrade connectors.
- Versions of the connectors you want to upgrade must be available on the Manager to which you are connected. Remote upgrade is available only in ESM 4.0 or later, and only for 4.0.2 or newer connectors.
- As a prerequisite to upgrading Connectors, both the Manager and the connector you want to upgrade must be running.
- If you are running a 32-bit version of a SmartConnector, you cannot convert this connector to the 64-bit version through upgrade. You must perform a new install of the 64-bit version of the SmartConnector to run the 64-bit implementation of the connector.

Local Upgrade

To locally upgrade a connector:

1. Stop the running connector and run the connector installer. The installer prompts you for the location to install the connector.
2. Select the location of the connector that you want to upgrade. The message "**Previous Version Found. Do you want to upgrade?**" appears.
3. Select the option to continue and upgrade the connector. The original installation is renamed by prefacing characters to the original folder name; the upgraded connector is installed in the location **\$ARCSIGHT_HOME\current**.

Remote Upgrade from ESM

Note: Only Windows, Linux, and Solaris platforms are supported for connector remote upgrade from the Console.

ESM not only provides the ability to centrally manage and configure connectors, but also to update them remotely. You can use the **Upgrade** command on the Console to upgrade to newer versions of connector software for managed devices.

The **Upgrade** command lets you launch, manage, and review the status of upgrades for all connectors. A failover mechanism launches connectors with previous versions if upgrades fail. All communication and upgrade processes between components (Console, Manager, and connectors) take place over secure connections.

The Console reflects current version information for all of your connectors.

Running Connectors

Connectors can be installed and run in **standalone** mode, as a Windows **service**, or as a UNIX **daemon**. If installed standalone, the connector must be started manually, and is not automatically active when a host is re-started. If installed as a Windows service or UNIX daemon, the connector runs automatically when the host is re-started. Admin privileges needed to install and run as a service on Windows platforms. See ["User Privileges When Installing \(UNIX only\)" on the next page](#) for instructions on using **root** or **non-root** user privileges when running as a Linux/UNIX daemon.

Caution: Some SmartConnectors require that you restart your system before configuration changes take effect.

Connectors for scanners present a special case. To run a scanner connector in interactive mode, run in standalone mode and **not** as a Windows service or UNIX daemon.

Run Connectors in Standalone Mode

To run all installed connectors on a particular host, open a command window, go to **ARCSIGHT_HOME/current/bin** and run:

```
arcsight connectors
```

To view the connector log, read the file:

```
$ARCSIGHT_HOME/current/logs/agent.log
```

To stop all connectors, enter **Ctrl+C** in the command window.

Tip: On Windows platforms, connectors also can be run using shortcuts and optional Start Menu entries.

Run Connectors as a Windows Service

Connectors installed as a service can be started and stopped manually using platform-specific procedures.

To start or stop connectors installed as services on Windows platforms:

1. Right-click **My Computer**, then select **Manage** from the **Context** menu.
2. Expand the **Services and Applications** folder and select **Services**.
3. Right-click on the connector service name and select **Start** to begin running the connector or **Stop** to stop running the service.

To verify that a connector service has started, view the file:

```
$ARCSIGHT_HOME/logs/agent.out.wrapper.log
```

To reconfigure a connector as a service, run the connector Configuration Wizard again. Open a command window on `$ARCSIGHT_HOME/current/bin` and run:

```
runagentsetup
```

See ["Install as a Service" on page 50](#) for further details.

Run Connectors as a UNIX Daemon

Connectors installed as a daemon can be started and stopped manually using platform-specific procedures.

On UNIX systems, when you configure a connector to run automatically, ArcSight creates a control script in the `/etc/init.d` directory. To start or stop a particular connector, find the control script and run it with either a **start** or **stop** command parameter.

For example:

```
/etc/init.d/arc_serviceName {start|stop}
```

To verify that a connector service has started, view the file:

```
$ARCSIGHT_HOME/logs/agent.out.wrapper.log
```

To reconfigure connectors as a daemon, run the connector Configuration Wizard again. Open a command window on `$ARCSIGHT_HOME/current/bin` and enter:

```
runagentsetup
```

See ["Install as a Service" on page 50](#) for further details.

User Privileges When Installing (UNIX only)

You can run a SmartConnector as a service or standalone. See ["Running Connectors" on the previous page](#) for more information.

SmartConnectors can be run as a **non-root** user, such as *arcsight*. A **SmartConnector** that listens to a port less than 1024 needs a **root** privilege to listen to a restricted port. For example, a syslog daemon connector needs a **root** privilege to bind to a restricted port such as port 514.

The following sections describe the recommended options for two concepts: 1) connectors that require to be configured to listen to low numbered ports; 2) connectors that are run as a service. Based on your specific installation and configuration, you may apply one or both concepts.

When Running As a Service

Option 1: This is the recommended option. Install as user *arcsight*, run as user *arcsight*.

The following instructions refers to user *arcsight* as a generic name for any user with **non-root** privileges.

When you log on as the user **arcsight** for installation, the ArcSight connector files will be owned by user **arcsight**.

Run as the user **arcsight** after installation to set up the connector wizard. There are a couple of items to note:

- If a Syslog Daemon connector is selected, then the configured port number must be 1024 or greater for this option (see ["Option 2: Install as user arcsight, run as user arcsight with port forwarding."](#) below).
- When running as a service, the setup wizard displays a dialog that states:

The Connector Setup Wizard is not able to modify the service configuration because the Wizard is not running as root. Please run this Wizard as root. Or to manually install, logged on as root, execute the following script:

```
$ARCSIGHT_HOME/current/bin/arcsight agentsvc -i -u user
```

To manually remove the service, logged on as root, execute the following script:

```
$ARCSIGHT_HOME/current/bin/arcsight agentsvc -r
```

We do not recommend to run the Wizard as **root**. Instead, run the Wizard as user **arcsight** and then manually install the service. Execute the following script while logged on as root to install the connector as a service:

```
$ARCSIGHT_HOME/current/bin/arcsight agentsvc -i -u arcsight
```

The **-u arcsight** option means that the service will be run as user **arcsight**.

Option 2: Install as user arcsight, run as user arcsight with port forwarding.

This option is the same as option #1 but it also allows a Syslog Daemon to receive events that are sent to ports below 1024. To use this option, use the same procedures as for option #1. In addition, use another program that forwards traffic from a low number port to the port configured for the connector. For example, if the syslog events are being sent to port 514 and the connector is configured to receive on port 6000, the forwarder re-routes from port 514 to port 6000. There are several possible programs that can do the port forwarding including **iptables**, **ncat**, and **socat**. The **iptables** program is packaged with some versions of Linux/Unix and is an easy choice.

Option 3: Install as user root, run as user root.

This option is less secure than the other options since **root** privileges are required for installation, configuration, and maintenance of the connectors.

A user logs on to the system as **root** and installs the ArcSight connector. This results in all of the ArcSight connector files to be owned by user **root**. The connector setup wizard is also run while logged on as **root**. If the connectors are to be run as a service, the service configuration is done by the connector setup wizard and no additional steps are required.

Caution: Avoid installing as user **arcsight**, and run as user **root**.

This can lead to security vulnerability issues. The potential problem with this option is that the connector configuration files are owned by user **arcsight** and so may be more susceptible to modification by a malicious user. Since the connectors are run as **root**, those modifications may result in undesirable privilege escalation.

When Running in Standalone Mode

Option 1: This is the recommended option. Install as user *arcsight*, run as user *arcsight*.

The following instructions refers to user **arcsight** as a generic name for any user with **non-root** privileges.

When you log on as the user **arcsight** for installation, the ArcSight connector files will be owned by user **arcsight**.

Run as the user **arcsight** after installation to set up the connector wizard.

If a Syslog Daemon connector is selected, then the configured port number must be 1024 or greater for this option (see ["Option 2: Install as user *arcsight*, run as user *arcsight* with port forwarding."](#) on the previous page).

Option 2: Install as user *arcsight*, run as user *arcsight* with port forwarding.

This option is the same as option #1 but it also allows a Syslog Daemon to receive events that are sent to ports below 1024. To use this option, use the same procedures as for option #1. In addition, use another program that forward traffic from a low number port to the port configured for the connector. For example, if the syslog events are being sent to port 514 and the connector is configured to receive on port 6000, the forwarder re-routes from port 514 to port 6000. There are several possible programs that can do the port forwarding including **iptables**, **ncat**, and **socat**. The **iptables** program is packaged with some versions of Linux/Unix and is an easy choice.

Caution: Avoid installing using the two following scenarios:

- as user **arcsight**, and run as user **root**

This can lead to security vulnerability issues. The potential problem with this option is that the connector configuration files are owned by user **arcsight** and so may be more susceptible to modification by a malicious user. Since the connectors are run as **root**, those modifications may result in undesirable privilege escalation.

- as user **root** and run as user **root**

This option is less secure since **root** privileges are required for installation, configuration, and maintenance of the connectors. A user logs on to the system as **root** and installs the ArcSight connector. This results in all of the ArcSight connector files to be owned by user **root**. The connector setup wizard is also run while logged on as **root**.

Verifying that a Connector is Running on ESM

To verify that a connector is running, you can check the ArcSight Console Navigator in the **Resources** tab, under **Connectors**. If the connector is running, you will see **<connector_name> (running)** listed.

Uninstalling a Connector

Before uninstalling a connector that is running as a service or daemon, first stop the service or daemon. Also, be sure to remove the service files using `$ARCSIGHT_HOME/current/bin/arcsight agentsvc -r` before uninstalling the connector.

The Uninstaller does not remove all the files and directories under the connector home folder. After completing the uninstall procedure, manually delete these folders.

To uninstall on Windows:

1. Open the **Start** menu.
2. Run the **Uninstall SmartConnectors** program found under **All Programs -> ArcSight SmartConnectors** (or the name you used for the folder during connector installation).
3. If connectors were not installed on the Start menu, locate the `$ARCSIGHT_HOME/current/UninstallerData` folder and run:
`Uninstall_ArcSightAgents.exe`

Note: To perform a silent uninstall, run the command with the following parameters:
`Uninstall_ArcSightAgents.exe -i silent`

To uninstall on UNIX hosts:

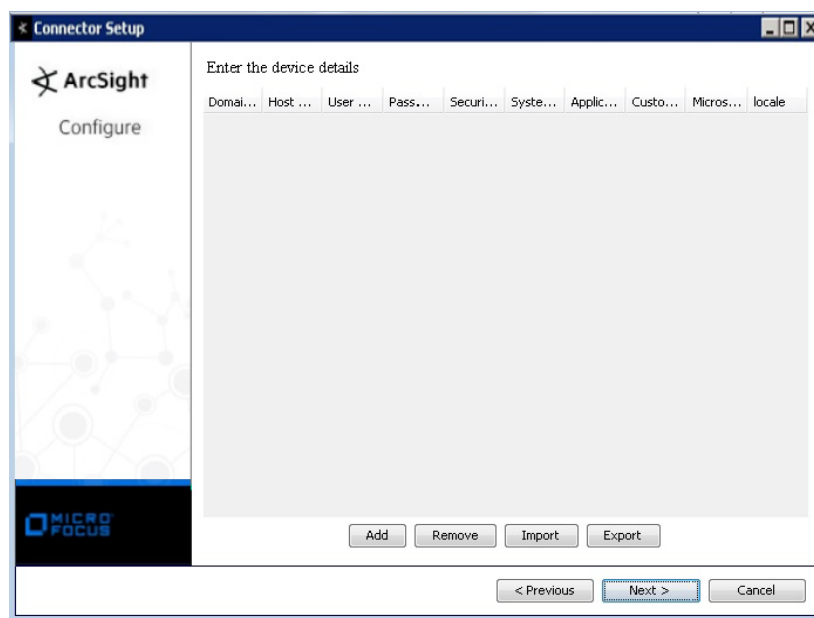
1. Change to the `$ARCSIGHT_HOME/UninstallerData` directory.
2. Run the command: `./Uninstall_ArcSightAgents`.

Note: The UninstallerData directory contains a file `.com.zerog.registry.xml` with Read, Write, and Execute permissions for everyone. On Windows platforms, these permissions are required for the uninstaller to work. However, on UNIX platforms, you can change the permissions to Read and Write for everyone (that is, 666).

Note: To perform a silent uninstall, run the command with the following parameters:
`./Uninstall_ArcSightAgents -i silent`

Working with the Windows Hosts Table

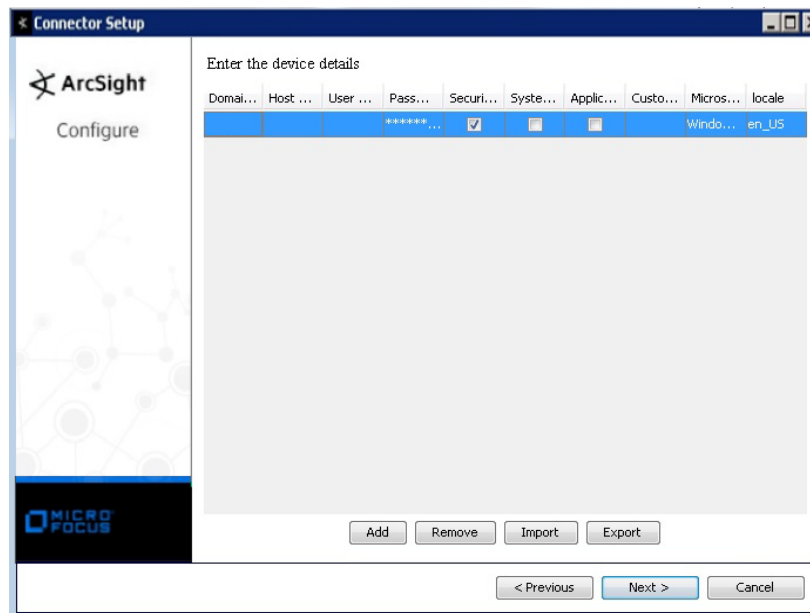
During connector installation, a connector using table parameters shows the following type of window for entering parameter data. Connectors for which parameter tables are used include multiple file, multiple site or server, and multiple database instance connectors.



The parameters for this type of connector can be entered manually for a few lines of data, or, for a larger number of entries, you can import a **.csv** file. You can also create a **.csv** file by exporting data you've already entered. See ["Importing and Exporting CSV Files" on the next page](#) for specific steps.

Manually Entering Table Parameter Values

To enter parameters manually, use the **Add** button to create fields and enter the data, as shown below.



If needed, use the **Export** button to export your parameter table data into an external **.csv** file to save for later use.

Note the following when using this feature:

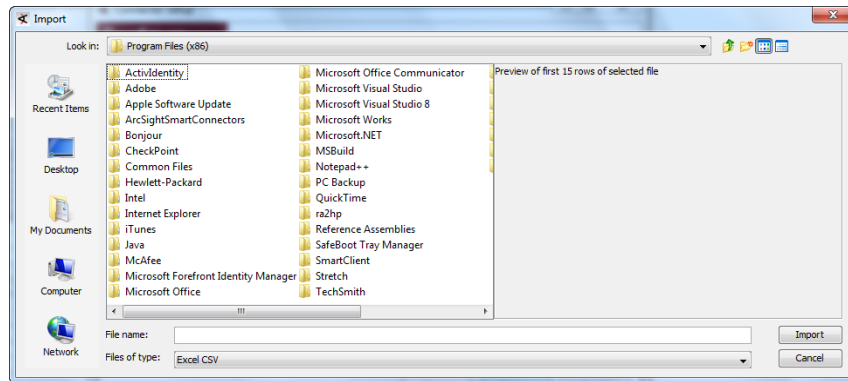
- Columns that contain private data (shown as asterisks), such as passwords, **will not appear in exported files** after using the **Export** button.
- After importing a **.csv** file (using the **Import** button), data in private columns remain hidden (shown as asterisks).
- Although you can manually enter a private column (either by adding the column to your CSV within a spreadsheet program or by filling it in through the Configuration Wizard), it still will not appear in any exported files. This is a precautionary measure.
- Importing data from a **.csv** file (using the **Import** button) causes all existing data in the table to be removed and replaced by the incoming data.

Importing and Exporting CSV Files

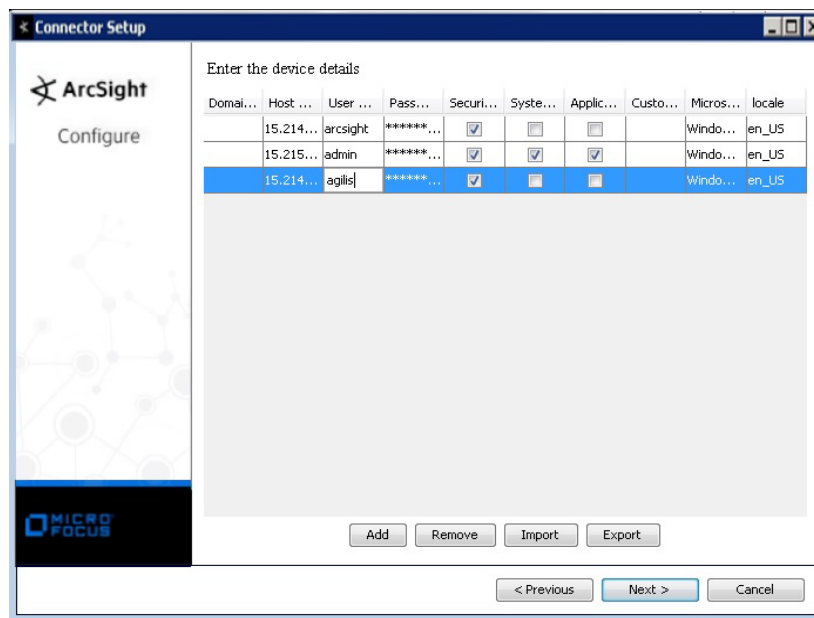
You can populate many lines of parameter data by creating a **.csv** file, then using the **Import** button to fill the parameter entry table of the Configuration Wizard.

To use the Import feature:

1. Using a spreadsheet program (such as Microsoft Excel), enter the parameter data into a table and save it as a **.csv** file.
2. During connector installation, click the **Import** button to locate the **.csv** file you created. The window previews the CSV file contents.



3. Click the **Import** button on the Import window. This populates the connector parameters fields.



4. You can add more rows manually (using the **Add** button) and then export the resulting table (using the **Export** button) to an external **.csv** file for later use.

Note: The example above shows a “Password” column within the Configuration Wizard that does not appear in the original **.csv** file. This private column does not contain actual password data and **will not be included in an exported file**.

5. If you are finished entering data, click **Next**.

Chapter 4: Configuring Connectors

Most of the sections in this chapter discuss configuration tasks you can perform without access to the Manager. The exception is ["Defining Default and Alternate Configurations from the ArcSight Console" on page 56.](#)

Modifying Connectors

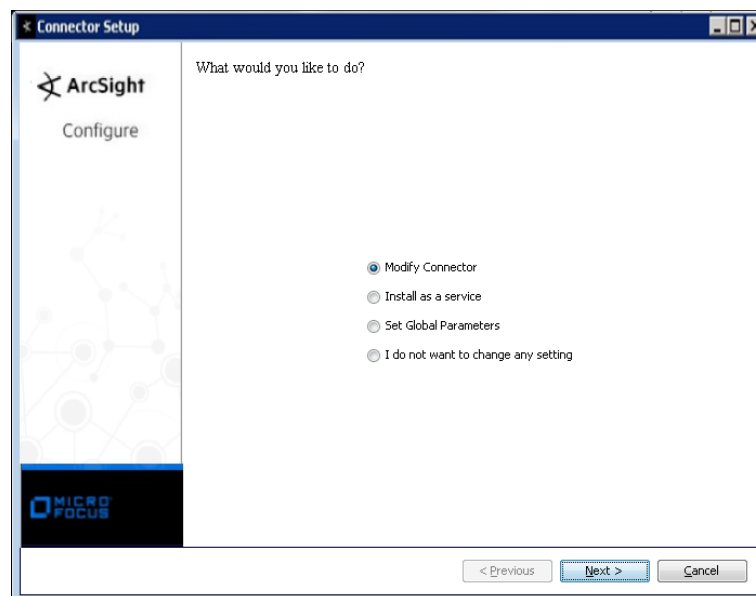
This section addresses modifying connectors parameters you initially configured through the wizard, including destination parameters, service settings, and setting global parameters.

To change configured settings:

After first installing a connector, you can run the wizard again if you want to modify settings. From `$ARCSIGHT_HOME/current/bin`, execute:

runagentsetup

The following window is displayed.



Modify Connector

To make changes to the initial values set during connector installation and configuration, select **Modify Connector**.

Modify Connector Parameters

The information shown in the windows in the steps that follow is meant as example data. Your windows will show different data, depending upon the connectors you have installed and their configuration.

To change parameter values:

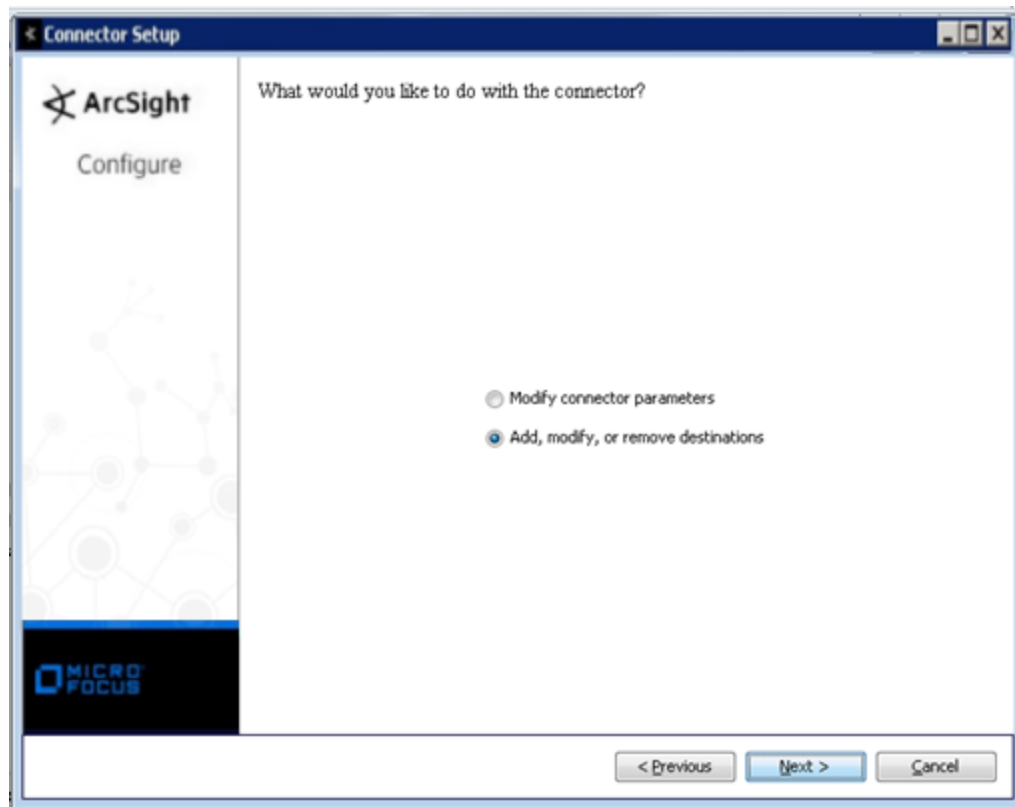
1. Once you have started the wizard and selected **Modify Connector**, after clicking **Next**, you can choose whether to **Modify connector parameters** or **Add, modify, or remove** destinations. Select **Modify connector parameters**.
2. Modify parameters as needed in the parameters windows displayed. The parameters shown will be specific to the connector you have installed.
3. Click **Next**. The connector parameters changes are processed and the connector configuration is modified. When the configuration changes are complete, you will receive the message **Successfully updated parameters**.
4. Click **Next**. Choose **Exit**, to complete the connector modification, or choose **Continue**, to continue to make connector modifications. Click **Next** to exit or continue.

Add, Modify, or Remove Destinations

Modify your existing destination or destinations or add a destination. Shows the destination or destinations configured during connector installation and configuration.

To add a destination:

1. After running the wizard, **Modify Connector** is selected by default. Do not change this selection.
2. Click **Next**. On the window displayed, select **Add, modify, or remove destinations**.
3. Click **Next**. The selections displayed depend upon the destination or destinations previously configured. You can modify the parameters and settings for these destinations, or you can select **Add destination** to add another destination.
4. Click **Next**; the window for adding, modifying, or removing destinations will be displayed.



To remove a destination:

1. After running the wizard, **Modify Connector** is selected by default. Do not change this selection.
2. Click **Next**. Select **Add, modify, or remove destinations**.
3. Click **Next**. From the list of destination selections, select the destination to remove.
4. Click **Next**. Select **Remove destination**.
5. Click **Next**. The destination removal is started.
6. Click **Next**. The destination removal is completed.
7. Click **Next**. Choose **Exit**, to complete the connector modification, or choose **Continue**, to continue to make connector modifications. Click **Next** to exit or continue.

Modify Destination Parameters

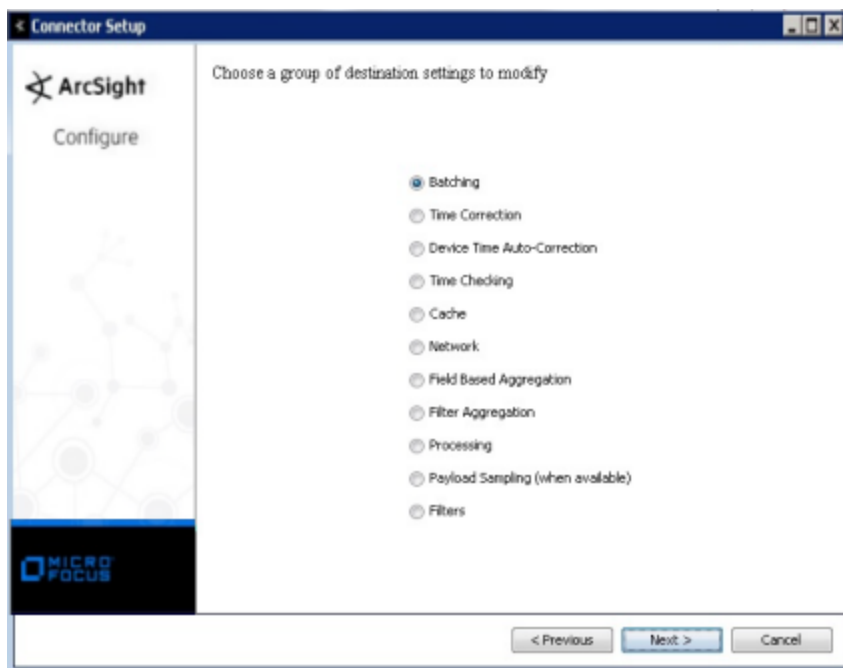
Select **Modify destination parameters** to modify values for the parameters set during initial destination configuration. The parameters displayed depend upon the connector previously configured.

Modify Destination Settings

ArcSight SmartConnectors can be configured to optimize their performance and increase their function. You can configure them to enable aggregation, batching, time correction, and payload sampling, as well as

specifying filtering conditions. Based upon filtering conditions, SmartConnectors can filter events sent to the selected destination.

1. After running the wizard, **Modify Connector** is selected by default. Do not change this selection.
2. Click **Next**. On the window displayed, select **Add, modify, or remove destinations**.
3. Make sure your destination is selected and click **Next**.
4. Select **Modify destination settings** to configure the following parameters:



For details about these parameters, see [Configuring Destination Settings](#).

Reregister Destination

When the Manager recognizes a connector, it generates an ID token the connector uses to identify its security events. If the Manager stops accepting events from a connector for an unknown reason, or if you have upgraded a connector but its resource was removed from the database, you may need to re-register the connector.

To reregister destination:

1. After running the wizard, **Modify Connector** is selected by default. Do not change this selection.
2. Click **Next**. Select **Add, modify, or remove destinations**.
3. Click **Next**. Select a current destination from the choices displayed. These vary depending upon initial connector configuration.
4. Click **Next**. Select **Reregister destination**.
5. Click **Next**. Enter any credentials required depending on the destination for the connector. The window is not displayed for destinations that do not require credentials.

6. Click **Next**. The reregistration begins.
7. Click **Next**. The reregistration completes.
8. Click **Next**. Choose **Exit** and click **Next**.
9. Restart the connector to apply the new ID token.

Add a Failover Destination

To add a failover destination:

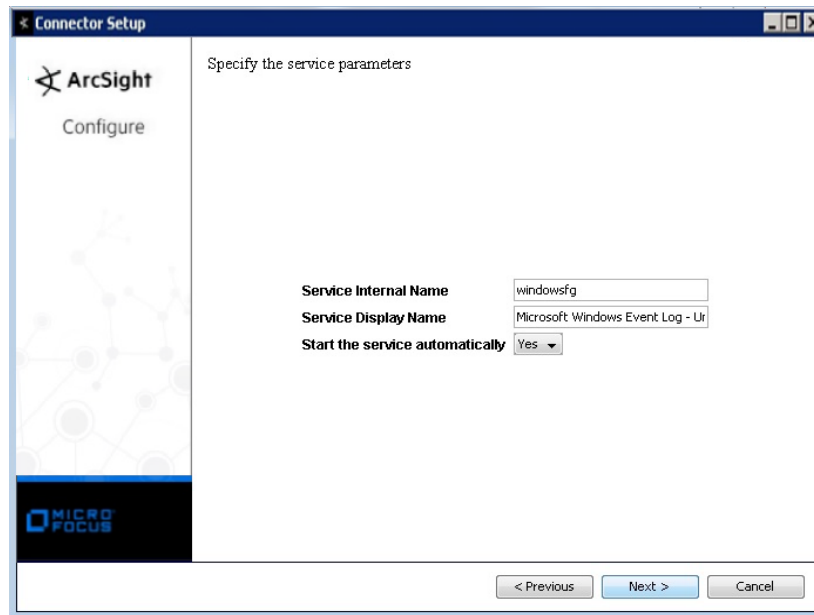
1. After starting the wizard, **Modify Connector** is selected by default. Do not change this selection.
2. Click **Next**. From the next window, select **Add, modify, or remove destinations**.
3. Click **Next**. Select a current destination from the choices displayed. These vary depending upon initial connector configuration.
4. Click **Next**. Select **Add fail a over destination**.
5. Click **Next**. Select the destination type.
6. Click **Next**. Enter the parameter settings for the failover destination.
7. Click **Next**. The destination parameter update begins.
8. Click **Next**. The destination parameter completes.
9. Click **Next**. Choose **Exit** and click **Next**.
10. To apply your changes, restart the connector.

Install as a Service

This section describes how to run a connector as a service, and how to remove a connector service.

To configure the connector to run as a service:

1. Once you have run the wizard, select **Install as a service**.
2. Click **Next**. Specify or change the service parameters.



3. Click **Next**. The service summary is displayed.
4. Click **Next**. Choose **Exit**, to complete the connector modification, or choose **Continue**, to continue to make connector modifications. Click **Next** to exit or continue.

To remove a SmartConnector service:

1. Select **Uninstall as a service**.
2. Click **Next**. Removal of the connector service is confirmed.
3. Click **Next**. Choose **Exit**, to complete the connector modification, or choose **Continue**, to continue to make connector modifications. Click **Next** to exit or continue.

Set Global Parameters

Select Set Global Parameters if you want to modify values for setting FIPS mode, remote management, or preferred IP version.

Connector Setup@n...

Please make your choice of settings

ArcSight
Configure

Unique Generator ID

FIPS Mode: Disabled

Remote Management: Disabled

Remote Management Listener Port: 9001

Preferred IP Version: IPv4

Format Preserving Encryption: Disabled

Format Preserving Host URL

Proxy Host (https)

Proxy Port

Format Preserving Identity

Format Preserving Secret

Event Fields to Encrypt: rawEvent,requestUrl,requestCookies,devic

< Previous Next > Cancel

Global Parameter	Setting
FIPS mode	Select to Enabled to enable FIPS compliant mode. To enable FIPS Suite B Mode, see Enable FIPS Suite B Mode 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 specified in Remote Management Listener Port 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 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.

If Format Preserving Encryption was left as **Disabled** during connector installation, you can set to **Enabled** and configure the other encryption-related parameters, shown below. If Format Preserving Encryption was set to **Enabled** during connector installation, encryption parameters cannot be modified. A fresh installation of the connector will be required to make any changes to encryption parameters.

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.

Global 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 Host 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.
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 authentication.
Event Fields to Encrypt	Recommended fields for encryption are listed; delete any fields you do not want encrypted from the list, and add any string or numeric fields you wish to be 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 the Add a Connector window. Continue the installation procedure with Select Connector and Add Parameter Information.

Additional Configuration

The following topics are additional configuration settings.

Enabling FIPS Suite B Mode

To enable FIPS Suite B Mode:

1. After completing installation, execute **runagentsetup** from the `$ARCSIGHT_HOME\current\bin` directory.
2. On the window displayed, select **Modify Connector**.
3. Select **Add, Modify**, or **remove destinations** and click **Next**.
4. Select the destination for which you want to enable FIPS Suite B mode and click **Next**.
5. Select **Modify destination parameters** and click **Next**.
6. When the parameter window is displayed, select **FIPS with Suite B 128-bits** or **FIPS with Suite B 192 bits** for the FIPS Cipher Suites parameter. Click **Next**.
7. The window displayed shows the editing changes to be made. Confirm and click **Next** to continue. (To adjust changes before confirming, click **Previous**.)
8. A summary of the configuration changes made is displayed. Click **Next** to continue.
9. Click **Exit** to exit the configuration wizard.

Specifying the IP Address on Devices with Multiple Network Interfaces

About

Devices with multiple network interfaces (multihomed host), let the connector choose the interface with the lowest numerical IP address to report its host address.

If the interface is not the one your device is using to communicate with another device, you may update the **connector.network.interface.name** property.

Procedure

To set this parameter:

1. After the connector installation, go to `$ARCSIGHT_HOME\current\user\agent`.
2. Go to the end of the `agent.properties` file and add the `connector.network.interface.name` parameter.
3. Enter the interface name.
4. Save the file.
5. Restart the connector.

Getting the Name of the Network Interface:

On an Linux Platform:

1. Run the command `ifconfig` in the terminal.

The following example shows interface names on the left column.

The interface names are `eth0` and `lo`.

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:94:37:b6
          inet addr:192.168.1.231  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe94:37b6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:519374 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110611 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:332864737 (332.8 MB)  TX bytes:11113451 (11.1 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:33651 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33651 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2951078 (2.9 MB)  TX bytes:2951078 (2.9 MB)
```

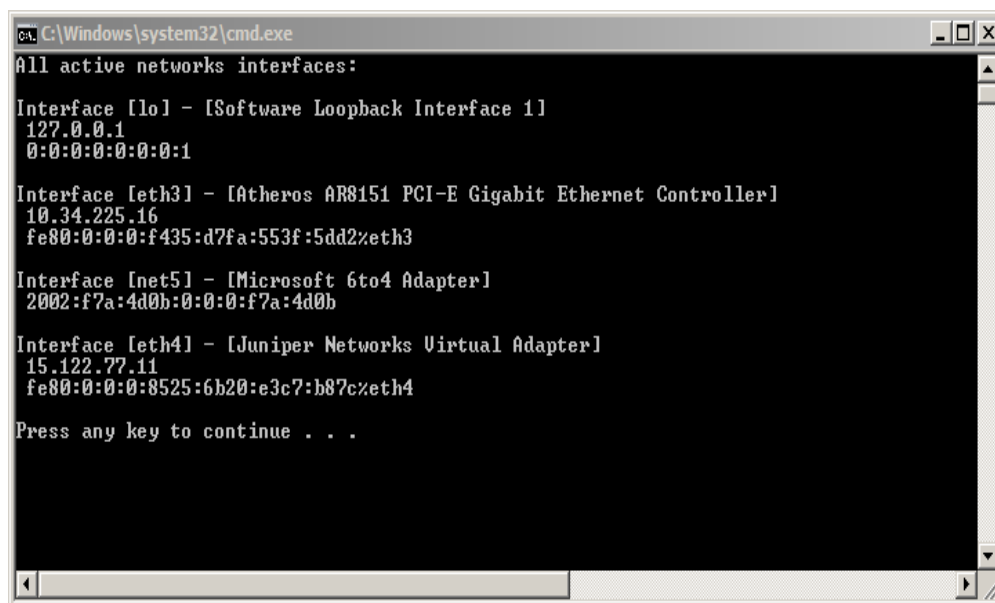
On a Windows Platform

The `ipconfig` command does not provide interface names, instead use this utility.

1. From `$ARCSIGHT_HOME\current\system\agent\config\list_net_interfaces`, run the `run.bat` file.

The following example shows the interface names inside the brackets.

The interface names are in the first brackets of each interface (lo, eth3, net5, eth4).



```
C:\Windows\system32\cmd.exe
All active networks interfaces:

Interface [lo] - [Software Loopback Interface 1]
127.0.0.1
0:0:0:0:0:0:1

Interface [eth3] - [Atheros AR8151 PCI-E Gigabit Ethernet Controller]
10.34.225.16
fe80:0:0:0:f435:d7fa:553f:5dd2%eth3

Interface [net5] - [Microsoft 6to4 Adapter]
2002:f7a:4d0b:0:0:0:f7a:4d0b

Interface [eth4] - [Juniper Networks Virtual Adapter]
15.122.77.11
fe80:0:0:0:8525:6b20:e3c7:b87c%eth4

Press any key to continue . . .
```

Lowering Network Bandwidth When Sending Events to ESM

Connectors can send event information to the Manager in a compressed format using HTTP compression. The compression technique used provides compression rates of 1 to 10 or greater, depending upon the input data (in this case, the events sent by the connector). Using compression lowers the overall network bandwidth used by connectors dramatically without impacting their overall performance.

By default, all connectors have compression enabled. To turn it off, add the following line to the **agent.properties** file (located at **ARCSIGHT_HOME\current\user\agent**):
http.transport.compressed = false

Defining Default and Alternate Configurations from the ArcSight Console

A SmartConnector can have a default and a number of alternate configurations.

An alternate configuration is a set of runtime parameters that is used instead of the default configuration during a specified portion of every day. For example, you might want to specify different batching schemes (by severity or size) for different times of a day. You can define more than one alternate configuration per destination and apply them to the destination for different time ranges during the day. For example, you can define a configuration for 8 am to 5 pm time range and another configuration for the 5 pm to 8 am time range.

To define default configurations:

1. In the Navigator panel, choose the **Connectors** resource tree.
2. In the Connectors resource tree, right-click the SmartConnector you want to manage and choose **Configure**.

This opens the **Inspect/Edit** panel for the **Connector Editor**. On the Connector tab, the Name field is automatically populated with the name assigned during SmartConnector Installation.

3. On the **Connector** tab, type the **Connector Location** and the **Device Location**. All events are tagged with these fields by the SmartConnector. Creation date and other information is automatically populated.
4. On the **Default** tab, change any additional Batching, Time Correction or other parameters as desired. See the *ArcSight Console User's Guide*, "Managing SmartConnectors", for configuration field descriptions in the "Connector Editor Option Tabs" and "Connector Tab Configuration Fields" sections.
5. Click **Apply** to add your changes and to keep the Connector Editor open.

The description entry associated with the setting provides tool tip information. These parameters are not localized since they come directly from the connector and the connector may contain new resources (since it may be a newer version).

The framework for connector commands operates in a similar way. Configuration of the connector command menu is achieved by sending the list of commands that are supported on the connector at registration time.

There are several controls you can adjust in the Connector Editor. The variety of options are best summarized by briefly describing what's available at each of the editor's tabs or subtabs.

To create alternate configurations:

1. Open the Inspect/Edit panel of the SmartConnector.
2. On the **Default** tab, click **Add Alternate**.
A new tab, Alternate #1, is added to the edit panel. The alternate tab provides fields for entering a time interval.
3. Under **Time Interval**, enter times for **From** and **To**. Make additional changes as required, then click **Apply**.
4. Repeat the process if you want additional alternates using different time intervals and different parameters. For example, create alternates if you want varying batching schemes based on severity or size on certain times of the day.

If the time ranges of the combined alternate configurations do not span 24 hours, the default parameters will be used to cover the time intervals not already defined in the alternates.

See "Managing SmartConnectors" in the *ArcSight Console User's Guide* for full details. For Management Center implementations, see the *ArcSight Management Center Administrator's Guide*, "Managing Alternate Configurations", for details.

Customized Events Filtering

Use customized events filtering to remove events that are not of interest, or include only the events that are of interest, to your organization before they are counted. Filtering is performed based on certain pre-defined patterns. All connector destinations subsequently receive only the relevant events based on the filtering defined.

By default, this feature is not enabled. If enabled, you can either include only the events that have a specific pattern in the raw event field, or exclude all the events that have a specific pattern. Use the [Get Status](#) command at any point in time when the connector is running to see:

- the total number of events filtered out since the last connector start
- the current status of the events filtering

Feature Usage

The filtering feature applies to the raw event field in the ArcSight security event. During the flow of the security events through the connector, the raw event field is extracted and evaluated to apply the filter.

To use the filtering feature, two out of the following three properties should be added to the **agent.properties** file. The first parameter must always be included plus one of the other two. They are:

```
customeventsfilter.regex.enabled=false  
customeventsfilter.regex.pattern.include=  
customeventsfilter.regex.pattern.exclude=
```

To apply filtering, set the first property to **true** and enter a valid regex pattern in **one** of the other two properties. There is no need to add these properties to the **agent.properties** file if you do not change them from their default values. See [Java Regex Patterns](#).

Note If the feature is enabled and both patterns are inadvertently defined, the exclude pattern takes precedence and the include pattern is ignored.

By default, the feature is disabled (**customeventsfilter.regex.enabled=false**) and no filtering is applied to any events.

Note: Enabling the filter through an include pattern filters out all the events in the raw event field that do not have the pattern in question. Therefore, be certain of the outcome that you want to achieve before enabling the include filter.

Note: All properties are considered unique to the agent. Therefore, avoid defining any property multiple times for either the include or exclude patterns.

All device events have the raw events field present when they reach the connector, and will be impacted by using this feature. Some internal events, such as **agent:017** (get status), also have the **rawEvent** field

present in the event and will be impacted by the filtering feature. Most of the internal events, such as **agent:030**, **agent:031**, or **agent:050** do not have the **rawEvent** field in the event and will not be impacted. This feature only impacts the events that have a non-empty **rawEvent** field.

In case you enable the feature but use an invalid or empty pattern on both include and exclude pattern fields, a [Get Status](#) command shows a message similar to the following for the filtering state:

Custom Filtering: Events Filtering State.....Events Filtering Disabled Due to Syntax Error in User Defined Regex

The following table shows the various states of the filter under different user entry combinations.

customeventsfilter.regex.enabled	customeventsfilter.regex.pattern.exclude	customeventsfilter.pattern.include	Result
false	Any pattern (valid, invalid, or empty)	Any Pattern (valid, invalid, or empty)	The filtering is disabled.
true	Valid and non-empty pattern	Any Pattern (valid, invalid, or empty)	The filtering is enabled with exclude filter. Include pattern has no impact.
true	Empty or invalid	Valid pattern	The filtering is enabled with include filter.
true	Empty or invalid	Empty or invalid	The filtering is disabled.

Java Regex Patterns

Use the information at the following link to learn about the details on how to use the JAVA regex patterns:

[Java 8 Pattern Class](#)

If a bad regex (un-compileable by JAVA Pattern class) is used, an error message is logged in the **agent.log** file. See [Log Messages](#).

Get Status

From the ESM Console

Use the **Get Status** command from the ESM Console to get the current filtering state and also the number of events filtered out by the feature since the last connector start.

In the ESM Console, right-click on the connector and select **Send Command > Status > Get Status**.

The command is sent to the connector and the result set is displayed. In the results, there will be two rows pertaining to the custom filtering feature. See the blue highlight in the following example:

The screenshot displays the SmartConnector application interface. On the left, the 'Navigator' panel shows a tree structure under 'Connectors' with 'testLoc' and 'syslog-main(running)' highlighted. On the right, the 'Viewer' panel shows the 'Connector Command' status for 'syslog-main'. The status includes a table with 'Time' (27 Mar 2017 11:31:16 PDT) and 'Connector' (syslog-main). Below the table, the status text reads: 'From Connector:: syslog-main (3Rt5jAVsBABDHRiscS3ggDQ==)', 'Status Generated: Mon Mar 27 11:31:16 PDT 2017', and 'Memory Usage: 32Mb out of 230Mb'. A detailed list of metrics follows, including 'Agent Type', 'Agent Version', 'CommandResponses Processed', 'Custom Filtering: Events Filtered Out', 'Event rate LTC', 'Events Processed', 'Events/Sec', 'FCP Version', 'FIPS Enabled', 'First CommandResponse Processed', 'First Event Processed', 'Host Address', 'Host Name', 'Last CommandResponse Processed', 'Last Event Processed', 'Parser AUP Version', 'Queue Drop Count', 'Queue Rate', and 'Queue Rate (SLC)'. The 'Custom Filtering: Events Filtering State' is highlighted in blue, showing 'Events Filtering Enabled Through Exclude Filter'.

From the Command Line

To get status from the connector command line, enter this command from the <ARCSIGHT_HOME>/current/bin:

```
arcsight agentcommand -c status
```

Examples of Patterns

Patterns are compiled through the `java.util.regex.Pattern` class. Any non-empty pattern that can be compiled is considered a valid pattern. Below are a few examples of valid patterns and their results:

Example of Valid Pattern	Result
<code>customeventsfilter.regex.pattern.exclude=IPSec\\s+tunnel</code>	Filters out all the events that have the pattern IPsec tunnel in the raw event.
<code>customeventsfilter.regex.pattern.exclude="Bad\\s+\\S+"</code>	Filters out all the events that have the pattern "Bad anyWord" in the raw event (including the double quotes).

Example of Valid Pattern	Result
<code>customeventsfilter.regex.pattern.exclude=111.112.113.114</code>	Filters out all the events that have the IP 111.112.113.114 in the raw event.
<code>customeventsfilter.regex.pattern.include=remote_peer-_ip\\s*\\s*\\d+\\.\\.\\d+\\.\\.\\d+\\.\\.\\d+</code>	The filtering feature is enabled (provided that the exclude pattern is empty) through the include filter to allow only the events that have the pattern, for example, remote_peer-_ip = 11.12.13.14 in the raw event to pass through.

The following 10 messages are actual raw events. Examples of how the filtering can be used to include or exclude events from these 10 raw events are provided in the four cases that follow this list.

1. Nov 28 22:03:21 10.0.111.2 Nov 28 2016 22:02:17: %PIX-6-106015: Deny TCP (no connection) from 101.102.103.104/3671 to 10.0.111.22/80 flags RST ACK on interface inside
2. Nov 28 22:03:21 10.0.111.2 Nov 28 2016 22:02:17: %PIX-2-106006: Deny inbound UDP from 10.0.65.116/2908 to 10.0.126.55/123 on interface outside
3. Nov 28 22:03:53 10.0.111.2 Nov 28 2016 22:02:49: %PIX-2-106020: Deny IP teardrop fragment (size = 32, offset = 0) from 101.102.103.104 to 10.0.126.55
4. Nov 28 22:04:09 10.0.111.2 Nov 28 2016 22:03:04: %PIX-2-106001: Inbound TCP connection denied from 10.0.65.116/3694 to 10.0.126.55/23 flags SYN on interface outside
5. Nov 28 22:04:10 10.0.111.2 Nov 28 2016 22:03:05: %PIX-3-305005: No translation group found for tcp src inside:10.0.112.9/37 dst outside:10.0.65.116/3562
6. Nov 28 22:04:44 10.0.111.2 Nov 28 2016 22:03:39: %PIX-2-106001: Inbound TCP connection denied from 10.11.12.13/3699 to 10.0.126.55/8080 flags SYN on interface outside
7. Nov 28 22:05:07 10.0.111.2 Nov 28 2016 22:04:02: %PIX-4-500004: Invalid transport field for protocol=17, from 10.0.142.116/1234 to 10.0.126.55/0
8. Nov 28 22:05:25 10.0.111.2 Nov 28 2016 22:04:20: %PIX-2-106020: Deny IP teardrop fragment (size = 36, offset = 0) from 10.11.12.13 to 10.0.126.55
9. Nov 28 22:06:01 10.0.111.2 Nov 28 2016 22:04:57: %PIX-2-106012: Deny IP from 10.0.142.116 to 10.0.126.55, IP options: "0x1f"
10. Nov 28 22:06:10 10.0.111.2 Nov 28 2016 22:05:05: %PIX-3-305005: No translation group found for tcp src inside:10.0.112.9/37 dst outside:101.102.103.104/3562

The following cases describe the results of four distinct filtering cases on the above raw events.

Case 1:

```
customeventsfilter.regex.enabled=true
customeventsfilter.regex.pattern.exclude=Deny IP.*from \\d+\\.\\.\\d+\\.\\.\\d+\\.\\.\\d+
```

Events #3, #8, and #9 will be dropped (excluded) from the flow. This pattern is meant to exclude all raw events that have both the patterns <Deny IP> and <from IPAddress> in the same raw event.

Case 2:

```
customeventsfilter.regex.enabled=true  
customeventsfilter.regex.pattern.exclude=(10.11.12.13)|(101.102.103.104)
```

Events #1, #3, #6, #8, and #10 will be dropped (excluded) from the flow. The pattern is meant to exclude raw events that have the IPs 10.11.12.13 or 101.102.103.104.

Case 3:

```
customeventsfilter.regex.enabled=true  
customeventsfilter.regex.pattern.include=(10.11.12.13)|(101.102.103.104)
```

Events #2, #4, #5, #7, and #9 will be dropped (excluded) from the flow. The pattern is meant to include raw events that have the IPs 10.11.12.13 and 101.102.103.104 in them (both IPs do not need to be in the same pattern). All other events that do not have either of the IPs will be dropped.

Case 4:

```
customeventsfilter.regex.enabled=false  
customeventsfilter.regex.pattern.include=(10.11.12.13)|(101.102.103.104)
```

No filtering will be done because the enabled property is false.

Log Messages in **agent.log**

During connector initialization, information and error messages regarding the filtering states and the patterns are printed in the **agent.log** file. The following lines are excerpts from the **agent.log** file. This shows an instance when the user defined an invalid regex in the exclude pattern:

```
[2017-03-24 16:07:54,485][INFO ][default.com.arcsight.agent.loadable._CustomEventsRegexFilter]  
[init] CustomEventsRegexFilter Initialized: Filtering Enabled =true, Exclude Regex =remote_peer_  
ip\s+\is\s+\d+\d+\d+\d+, Include Regex =  
  
[2017-03-24 16:07:54,485][ERROR][default.com.arcsight.agent.loadable._CustomEventsRegexFilter]  
[init] Unable to compile custom filter exclude regex=remote_peer_ip\s+\is\s+\d+\d+\d+\d+  
  
[2017-03-24 16:07:54,500][INFO ][default.com.arcsight.agent.loadable._CustomEventsRegexFilter]  
[init] Events Filtering Disabled Due to Syntax Error in User Defined Regex
```

Connector Network and Zone Configuration

Network Model: Networks and Zones

A network model is a set of networks with its particular zones. The zones, unlike the networks, do not overlap. Since destinations can have different network models, a connector can set specific zones when sending to multiple destinations. Consequently, the network model data for each destination must be kept separately as well as the components in the destination-specific event flow.

The network model comes from:

- 3.5 or later ESM : used for ESM destinations, non-ESM destinations (if there is an ESM and the AUP Master feature is in use).
- An ArcMC-pushed Network Model.

The `networks.csv` and `zones.csv` files described below are for the latter option (ArcMC). For ESM network models, see the *Micro Focus Security ArcSight ESM Administrator's Guide* available at: <https://community.softwaregrp.com/t5/ESM-and-ESM-Express/tkb-p/esm/label-name/esm%206.11.0>.

Networks CSV:

The `networks.csv` file defines the networks used in the `zones.csv` file.

Tip: While creating zones, enter the exact URI for each network. Any zones connected to unknown networks (or most likely to networks incorrectly specified) cannot be used.

Add the Network:

1. Add the header, like in the following example: **#Type,Name,Parent Group URI,Customer URI** with no extra spaces. The code only tolerates a difference in upper and lower cases.
 - a. First column: Name it **"Global"** or **"Network"**. **Global** must appear at least once. The Network field is used to define other networks.
 - b. Second column: Add a name for each network.
 - c. Third column: Name the URI of the parent group in the network URI hierarchy.
 - d. Fourth column: (Optional) When no additional information is entered, add a comma after the parent group URI, then all the rows can show four columns. This column must be blank for lines with Global in the first column, and can be blank for lines with Network in the first column. If the field is not blank, the value is displayed in the customer's URI field. Any events that do not have this specific value in the customer's URI field are not considered by the network (or the zones within it).

The order of the networks depends on how the non- global (user) networks are displayed.

Zones CSV:

The **zones.csv file** defines the zones within the networks already populated in the **networks.csv file**. The file is divided into 5 columns and each row defines a zone. Add the Zones:

1. Add the header, like in the following example: **#Name,Start Address,End Address,Parent Group URI,Network URI** with no extra spaces. The code only tolerates a difference in upper and lower cases.
 - a. First column: Define the name of the zone.
 - b. Second column: Name the starting IP address in the range.
 - c. Third column: Name the ending IP address in the range. It can either IPv4 or IPv6 as long as it the same type of the starting IP address in the previous column.
 - d. Fourth column: Label the URI of the parent group in the zone URI hierarchy. This is the first part of the zone URI.
 - e. Fifth column: Add the network URI to define its corresponding zone. It must be an exact match, the concatenated URI from a network defined in the **networks.csv file** (the third column and second column, concatenated, in one row of the file).

Note: Events with addresses that fall outside the zones defined by the network model cannot be changed, so defining a network model does not necessarily affect all events.

To discard incoming zones and apply new ones (where applicable), go to Destination Settings < Network Group < Population Mode and set your SmartConnector to "Rezone (override)" instead of "Normal". This action may also be done from ArcMC.

Debugging tips:

- From your destination, search for **AddrBasedSysZonePopRows** and **AddrBasedUsrZonePopRows** in **get status results** or check the logs in **logStatus**. Look for errors and consider the time range in which the network model was pushed, or when the connector was started.

For additional information on ArcMC network models, go to Chapter 7: Managing Configuration of *Micro Focus Security ArcSight Management Center Administrator's Guide* available at:

<https://community.softwaregrp.com/t5/ArcSight-Management-Center-ArcMC/tkb-p/arcsight-management-center>.

Global Event ID Feature

About

Global Event ID is an optional feature that can be enabled by updating certain parameters. Ideally, each event passing through an ArcSight product should be assigned a Global Event ID.

Global Event IDs follow a sequential order that can register up to one million instances per second. Previous SmartConnector versions must be upgraded so that the events are properly assigned with Global Event IDs. Global Event IDs cannot be unassigned.

Global Event IDs cannot be encrypted.

Global Event ID Parameters:

Users can assign a Unique Generator ID while setting up the SmartConnector in silent mode. The new parameter 'gi' differences the Generator ID from the one in the silent template:

```
arcsight agentsetup -c -i silent -f <silent template> -gi <generator id>
```

When not provided, connectors, simply do not have a Generator ID and the Global Event ID is zero.

Unique Generator ID

A unique Generator ID is required for connectors to generate unique Global Event IDs.

The valid Generator ID value is 1 to 16383.

Note: Internal and audit events may be generated in a destination or in a connector. Therefore, each of them are unique events with unique Global Event IDs. If an internal or audit event is duplicated, the Global Event ID and the Event ID will be the same.

Procedure

Updating a Unique Generator ID

1. Go to agent.properties and enter agent.generator.id=<unique_generator_id>

Enabling the Global Event ID Feature

To enable the Global Event ID Feature:

Fresh Standalone Installation

1. Set the Generator ID. From Global Parameters, go to the field **Unique Generator ID** and enter a valid value.

The wizard continues with expected connector configuration.

If a value is not entered:

- a. The Global Events ID generated by the connector sets zero as the default value.
- b. The connector wizard displays a message, indicating that the Generator ID has not been set.
- c. The agent.log displays a message, indicating that the Generator ID has not been set.

ArcMC Managed Connectors

ArcMC can automatically assign a Generator ID to the connector, for more information, see the ***ArcSight Management Center Administrator's*** Guide.

If the ArcMC is not configured to automatically create a Generator ID, users can manually add the Generator ID.

1. Before adding the connector to the ArcMC, go to agent.properties and enter valid value.

Follow this format: agent.generator.id= <generator_id>.

Upgrading the Connector

Local Upgrade

1. Add the property agent.generator.id. Go to agent.properties and enter a valid Unique Generator ID.
Follow this format: agent.generator.id= <generator_id>

Remote Upgrade from ESM

When upgrading from ESM a SmartConnector version that does not support Global Event ID to a version that supports the feature, users may decide whether to enable the Global Event ID feature or not. If yes;

1. Before the upgrade, from agent.properties, enter agent.generator.id= <generator_id>
If the property is not entered, a message is displayed, indicating that the Global Event ID is zero.

Remote Upgrade from ArcMC

When upgrading from ArcMC, a SmartConnector version that does not support Global Event ID to a version that supports the feature, users may decide whether to enable the Global Event ID feature or not. If yes:

1. ArcMC can automatically assign a Generator ID to the connector, for more information, see the ***ArcSight Management Center Administrator's*** Guide.

2. In case your ArcMC is not supporting Global Event ID yet, go to `agent.properties`, enter `agent.generator.id=<generator_id>`.

Chapter 5: Connectors with Management Center

ArcSight produced two solutions for the central management of multiple connectors: Connector Appliance and ArcSight Management Center. Connector Appliance is an ArcSight legacy product that enabled central management and monitoring of multiple connectors. Its successor, ArcSight Management Center (Management Center) includes all of the Connector Appliance management functionality, but its capabilities also include management and monitoring of an additional range of ArcSight products, such as Loggers and other ArcSight Management Centers. Management Center features a web-based user interface to enable the management of local or remote connectors.

For specific information on the operation of Management Center, see the *Micro Focus ArcSight Management Center Administrator's Guide*.

Connectors that forward events to ESM can be managed using the Console, so Management Center is not required if all connectors have ESM as their only destination. However, Management Center is very useful when connectors target multiple heterogeneous destinations (for example, when Logger is deployed along with ESM), in a Logger-only environment, or when a large number of connectors are involved, such as in a MSSP deployment.

Management Center connectors are grouped in **containers**. Each container is a Java Virtual Machine (JVM) that can contain multiple connectors.

Managing Connectors on Management Center

Management Center manages three types of connectors:

- ["Local \(on-board\) Connectors" below](#)
- ["Remote Management Center Connectors" on the next page](#)
- ["Software-Based Connectors" on the next page](#)

Local (on-board) Connectors

Management Center includes multiple containers and on-board connectors. The manager interface can be used to manage these local connectors as well as remote connectors.

Note: Busy on-board connectors may impact the performance of the Management Center web-based interface.

Remote Management Center Connectors

Management Center can manage connectors on remote Management Centers, as well as other ArcSight hardware solutions such as Logger.

Software-Based Connectors

Previously-installed, software-based connectors can be remotely managed by some Management Center models, but the remote management feature is disabled on software connectors by default.

Note: You do not need to do the following processes for ESM or Express. These processes are only done for SmartConnectors running as a service, not for standalone SmartConnectors because they cannot be restarted automatically.

To manage software-based connectors with Management Center, you need to enable remote management on them. Add the following property to the **user/agent/agent.properties** file in the installation directory of each connector that you want to manage with Management Center:

remote.management.enabled=true

Restart the connector for property changes to take effect.

You can also customize the port on which the connector will be listening. By default, this port is set to 9001, but it can be changed by adding the following property to **user/agent/agent.properties**:

remote.management.listener.port=9002

In the example above, the connector listens on port 9002.

Caution: Only fifth-generation connectors support remote management, so you will need connector build 4855 (4.0.5.4878.0) or later to use this feature. Remote Management is not supported on connectors running AIX. This limitation is due to elements within the AIX platform.

Tip: Multiple software-based connectors installed on the same host require a separate port assignment. The default port for connectors is **9001**, so the second connector installed on the same host should use an alternate port. Micro Focus recommends using port **9002, 9003, 9004**, and so on.

For a complete list of all connectors supported by Management Center, see the Management Center Release Notes. You can also visit the Community site at <https://community.softwaregrp.com/t5/ArcSight/ct-p/arcSight>. ArcSight adds new connectors regularly.

Login Credentials for Software-Based Connector Remote Management

Login credentials are required for software-based connector remote management. Each connector ships with default credentials, which are provided below. The username cannot be changed. To change the default password, administrators can refer to "Changing Container Credentials" in the *ArcSight Management Center Administrator's Guide*.

Note: Load Balancer only works with connectors that use default remote management user name and password values.

Verify with your administrator what are the correct credentials for your environment.

The default connector remote management credentials are:

- Username: **connector_user**
- Password: **change_me**

Choosing a Deployment Scenario

Management Center can be deployed wherever connectors are needed, providing the following benefits:

- Connector management without ESM (that is, Logger-only environments)
- Remote control of runtime parameters, such as bandwidth control
- Centralized connector upgrade management and control
- Centralized troubleshooting of specific connectors

ArcSight Logger

Logger receives events from and sends to connectors, but lacks the depth of connector management found in ESM.

A Logger-only deployment benefits from Management Center in many ways, and provides most, but not all, of ESM's management function (for example, it does not contain the filter designer). Management Center also offers features that ESM does not, such as bulk operations (enabling control of many connectors at one time).

Management Center also can configure connectors with failover destinations, providing central failover control when redundant Loggers are deployed for this purpose. All or some connectors can be configured to send events to a second Logger or to an event file in the case of communication failure with the primary destination.

For more detailed information about Logger, see ["ArcSight Logger SmartMessage \(encrypted\) Destination" on page 91](#)

ArcSight ESM

Deploying Management Center in an ESM environment centralizes connector upgrade, log management, and other configuration issues. For more information, see ["ArcSight Manager \(encrypted\)" on page 76](#).

ESM and Logger

Management Center centralizes control when events are sent to ESM and Logger simultaneously. In one scenario, all events are sent to Logger while only high-value events are sent to ESM (for further analysis, for example). In another scenario, all events are sent to both, but Logger implements a longer retention policy.

Although each connector has specific destination parameters, Management Center allows for “bulk” management, eliminating the need to manually access each remote connector host to add or change destinations.

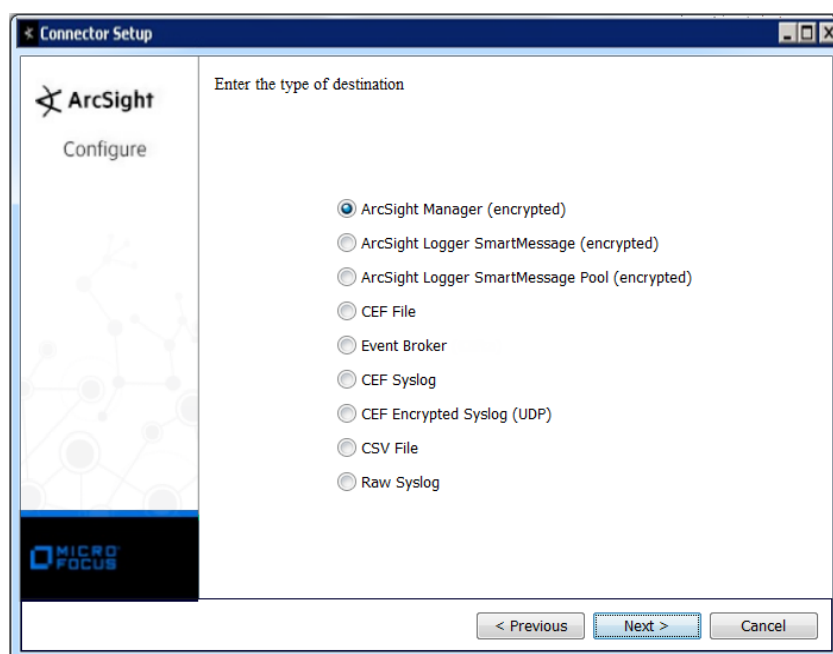
For detailed information and instructions for using Management Center, see the ***Micro Focus ArcSight Management Center Administrator's Guide***.

Chapter 6: Connector Destinations Overview

This chapter provides information about configuring a connector to send events to one or more destinations. A destination is a Manager or device that can receive events from a particular connector. In addition to the selections displayed during connector configuration explained below, events can be sent to additional or failover destinations.

Connector Destinations

During connector installation, you are asked to select a destination for the events collected by the connector. The following window shows the destination selections:



ArcSight Manager (encrypted)

This is the main destination used. When connectors send events to a Manager, the Manager stores the events in a relational database, processes them using its correlation engine, and makes them visible to the Console or Web interfaces. See "[ArcSight Manager \(encrypted\)](#)" on page 76 and the Online Console Help for complete information.

For instructions about setting up FIPS with ESM and SmartConnectors, see **Configuring FIPS for ESM and SmartConnectors** on <https://community.softwaregrp.com/t5/ArcSight-Knowledge-Base-Articles/Configuring-FIPS-for-ESM-and-SmartConnectors/ta-p/1588695>.

ArcSight Logger SmartMessage (encrypted)

Connectors can send CEF events to Logger using an encrypted, optionally compressed channel called SmartMessage. Logger also can receive CEF syslog events from connectors. For more information, see ["ArcSight Logger SmartMessage \(encrypted\) Destination" on page 91.](#)

ArcSight Logger SmartMessage Pool (encrypted)

You can specify a pool of logger devices as a single destination while the events are distributed among the loggers in the pool. Each "Batch" of events processed by the connector is sent to the next logger in the pool in a round-robin fashion. See ["Configuring a Logger Pool Destination" on page 98](#) for more information.

CEF File

This selection allows you to capture security events in a Common Event Format (CEF) file rather than forwarding them to a Manager.

For more detailed information, see ["CEF Destinations" on page 102.](#)

Transformation Hub

This selection sends events in Common Event Format (CEF) or binary to an Transformation Hub topic. Once events are in Transformation Hub, any number of applications can retrieve them.

The **AUP Master Destination** and **Filter Out All Events** should be set to **True** for ESM. See [ArcSight Manager \(Encrypted\)](#).

For instructions about setting up FIPS with Transformation Hub and SmartConnectors, see [Configuring FIPS for Transformation Hub and SmartConnectors](#) on [Configuring FIPS for Transformation Hub and SmartConnectors](#).

For instructions about configuring TLS between Transformation Hub and SmartConnector, see [Configuring TLS between Transformation Hub and SmartConnector](#) on <https://community.softwaregrp.com/t5/ArcSight-Connectors/tkb-p/connector-documentation>.

For more detailed information, see ["Transformation Hub" on page 103.](#)

CEF Syslog

This selection sends events in Common Event Format (CEF) (converted to bytes using the UTF-8 character encoding), and provides three protocol options: UDP, TCP, and TLS.

TCP and **UDP** can be used to send to Logger (TLS cannot be used for this purpose). Data received using these protocols are received using a TCP or UDP Receiver. One such receiver can receive from more than one connector. TCP and UDP can also be used to send to a Syslog Daemon connector.

The **TLS** protocol establishes a secure channel and allows for one-way or two-way authentication. If the TLS protocol is chosen, the events can be received by the Syslog NG Connector.

For more details about this destination, see ["CEF Destinations" on page 102](#). For more details regarding the Syslog NG Connector, see the *SmartConnector for Syslog NG Daemon*.

CEF Encrypted Syslog (UDP)

This destination sends events in Common Event Format (CEF) through the UDP protocol, providing symmetric-key encryption. This option allows for a "Shared Secret" key that requires configuration to encrypt the data. This data can be decrypted on the receiver side by the CEF Encrypted Syslog (UDP) connector.

For more information on this destination, see ["CEF Destinations" on page 102](#). For more details on how to decrypt the data, see the *SmartConnector for ArcSight CEF Encrypted Syslog (UDP)*.

CSV File

This selection lets you capture events a connector normally would send to the Manager into a CSV file. This is an advanced topic; typical ArcSight configurations do not require the use of external files to communicate events to the Manager. For more information, see ["CSV File Destination" on page 111](#).

Raw Syslog

This destination sends raw syslog events through the UDP, TCP, or TLS protocol. This destination is used with the connector for Raw Syslog Daemon to collect raw, unparsed events for further processing. For more detailed information, see ["Raw Syslog Destination" on page 114](#). If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the syslog data (for source and timestamp).

Note: Connections to Qualys Cloud Platform require TLS 1.1 or higher.

Appendix 6: Add Destinations

Connectors send a copy of events to each additional destination for which it is configured. Additional destinations can be useful, for example, when you have a development ArcSight environment working in parallel with your production environment and you want to test rules and reports. You can configure multiple destinations and also have failover destinations for when the primary destination is unavailable.

In such cases, you can configure the connector to send alerts to both your production Manager and your development Manager to be able to view real-time event flows on both systems. Because the destinations are independent, you do not compromise the events sent to the production Manager. For more information on how to add, refer to ["Add, Modify, or Remove Destinations" on page 47](#).

Failover Destinations

Each connector destination can have a failover destination that receives security events from the connector for which it is configured. The failover activates when the primary destination (such as an Manager) is not available (as when a network problem occurs) or is not keeping up with incoming events. These events are backed up to the failover destination. The connector also, when possible, caches the events and resends them to the primary destination when flow is restored.

A failover destination is not active when the primary destination is available, so the reports and replay features within the secondary Manager could contain incomplete information. This feature performs as a real-time alternative for severe problems with the primary destination. Refer to [Add a Failover](#) for more information.

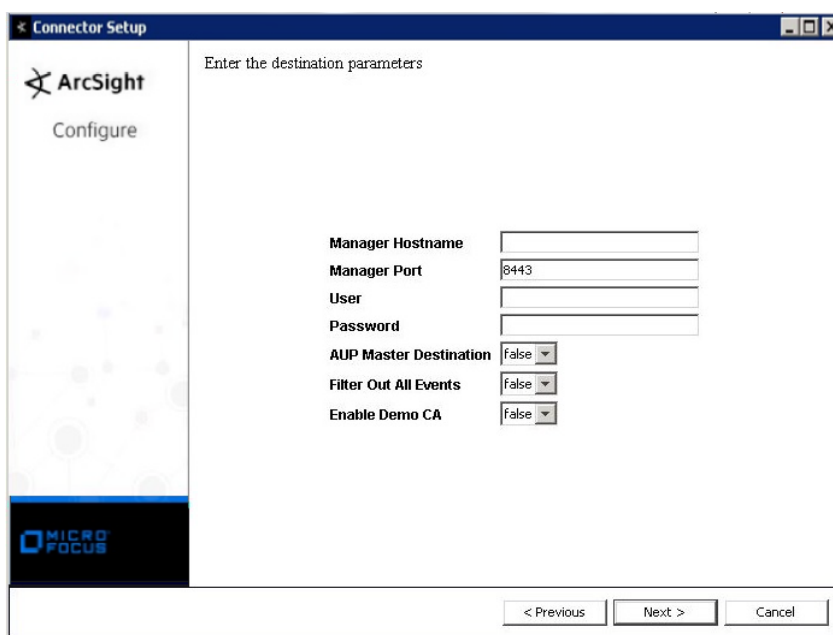
Chapter 7: ArcSight Manager Destination

This chapter describes the ArcSight Manager (encrypted) destination.

ArcSight Manager (encrypted)

When connectors send events to an ESM Manager, the ESM Manager stores the events in a relational database, processes them using its correlation engine, and makes them visible to the Console or Web interfaces.

1. Select a destination for the destination you are adding. See ["Connector Destinations" on page 72](#) to view the options.
2. Click **Next** to enter the destination parameters.



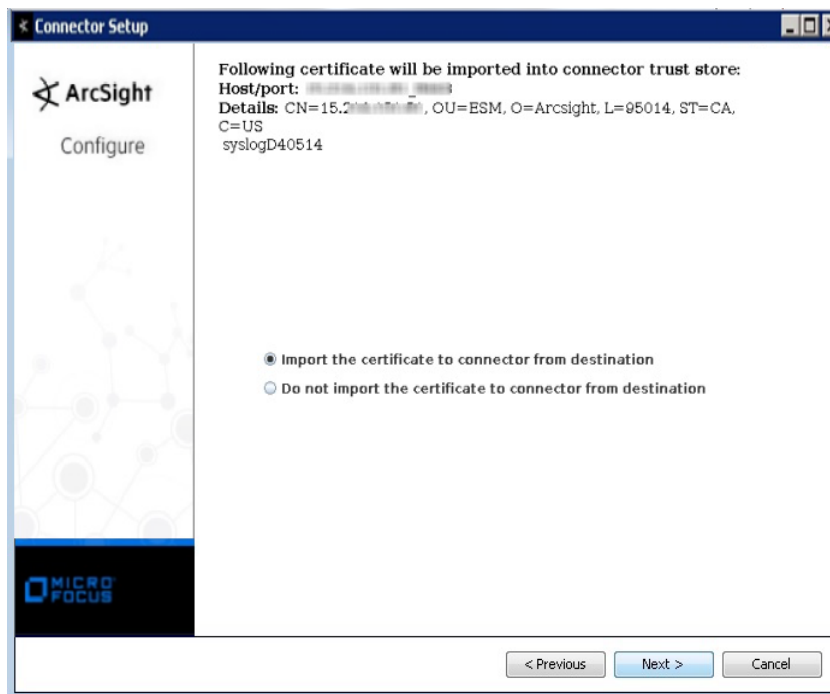
The screenshot shows a window titled "Connector Setup" with a sidebar on the left containing the ArcSight logo and the word "Configure". The main area is titled "Enter the destination parameters" and contains the following fields:

- Manager Hostname:
- Manager Port:
- User:
- Password:
- AUP Master Destination:
- Filter Out All Events:
- Enable Demo CA:

At the bottom of the window are three buttons: "< Previous", "Next >", and "Cancel".

Parameter	Description
Manager Hostname	<p>This is the local host name, IP address, or fully-qualified domain name of the machine where the ArcSight Manager is installed. This name is what all clients (such as ArcSight Console) specify to talk to the Manager. Using a host name and especially a fully-qualified domain name instead of an IP address is recommended for flexibility.</p> <p>The Manager host name is used to generate a self-signed certificate. The Common Name (CN) in the certificate is the Manager host name that you specify in this screen. Although the Manager uses a self-signed certificate by default, you can switch to using a CA signed certificate if needed. See the <i>ESM Administrator's Guide</i> for more information.</p>
Manager Port	8443
User	Enter a valid ESM User name.
Password	Enter the password for the ESM user.
AUP Master Destination	<p>Default: false. A connector can send events to ESM and non-ESM destinations simultaneously. In this configuration, it is helpful to use the AUP Master Destination feature. See ArcSight Content AUPs for more information.</p> <p>Note: Set this to True for ESM to use zone information from the Manager for non-Manager destinations, such as SmartMessage (Logger) or Transformation Hub.</p>
Filter Out All Events	<p>Default: false. If AUP Master Destination is set to true, you may or may not want to send this connector's events to that Manager. If the Manager should not get the events, set this to true. In that case the manager will only be used as a source of zone information. An example of when this would be a useful case is if the connector is sending events to the Transformation Hub, and ESM is reading those events from there.</p>
Enable Demo CA	<p>Default: false</p> <p>The ArcSight Manager host name is used to generate a self-signed certificate during ArcSight ESM installation. The Common Name (CN) in the certificate is the Manager host name that you specified during ESM installation.</p> <p>Do not use demo SSL certificates in production. Make sure when switching that you remove the demo CA from cacerts on all SmartConnectors and ArcSight Consoles.</p>

- Click **Next**. You will see the Performing add destination details.
- Click **Next** to continue.
- 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 connector from destination, the connector installation will end.)



6. You will see the dialog box with information about the connector(s) that have been updated and the primary destination. Click **Next** to continue.
7. Click **Exit** to complete the installation.

Chapter 8: Configuring Destination Settings

After configuring SmartConnector to send events, you can configure their operation further through the settings listed in Modify Destination Settings section. The details for the selections are provided in the following tables.

The following table shows the configurable settings.

Configurable Settings

Name Field	Value Field
Batching	SmartConnectors can batch events to increase performance and optimize network bandwidth. When activated, SmartConnectors create blocks of events and send them when they either (1) reach a certain size or (2) the time window expires. You can also prioritize batches by severity, forcing the SmartConnector to send the highest-severity event batches first and the lowest-severity event batches later.
Enable Batching (per event)	Create batches of events of this specified size (100, 200, 300, 400, 500, or 600 events). The default is 100. Caution: You could potentially lose data with batch sizes 500 and 600. Contact Customer Support before using 500 or 600 batch size.
Enable Batching (in seconds)	The SmartConnector sends the events if this time window expires (1, 5, 10, 15, 30, 60). Default is 5.
Batch By	This is Time Based if the SmartConnector should send batches as they arrive (the default) or Severity Based if the SmartConnector should send batches based on severity (batches of Highest Severity events sent first).
Time Correction	The settings in this group provide several ways to fix problems with devices that do not report the time correctly.
Use Connector Time as Device Time	(No Yes) Override the time the device reports and instead use the time at which the connector received the event. This option assumes that the connector is more likely to report the correct time. Default is No.
Enable Device Time Correction (in seconds)	The SmartConnector can adjust the time reported by the deviceReceiptTime field, using this setting. This is useful when a remote device's clock isn't synchronized with the ArcSight Manager. This should be a temporary setting. The recommended way to synchronize clocks between Manager and devices is the NTP protocol. This parameter also affects the startTime and endTime fields. Default is 0.
Enable Connector Time Correction (in seconds)	The SmartConnector can also adjust the time reported by the Connector Time SmartConnector itself, using this setting. This is for informational purposes only and lets you to modify the local time on the SmartConnector. This should be a temporary setting. The recommended way to synchronize clocks between Manager and SmartConnectors is the NTP protocol. Default is 0.

Configurable Settings, continued

Name Field	Value Field
Set Device Time Zone To	(Disabled <TimeZone>)(Default is Disabled) Ordinarily, it is presumed that the original device is reporting its time zone along with its time. And if not, it is then presumed that the SmartConnector is doing so. If this is not true, or the device isn't reporting correctly, you can switch this option from Disabled to GMT or to a particular world time zone. Select from the options available in the drop-down list. That zone is then applied to the time reported.
Device Time Auto-correction	Select from the time spans to do device-time auto-correction.
Future Threshold	The connector auto-corrects if the detect time is greater than the connector time by Future Threshold seconds. If either or both of the future and past thresholds are negative, auto-correction is disabled. Default is -1.
Past Threshold	The connector auto-corrects if the detect time is earlier than the connector time by Past Threshold seconds. Default is -1.
Device List	A comma-separated list of the devices to which the thresholds apply. The default, (ALL) means all devices.
Time Checking	
Future Threshold	The number of seconds by which to extend the connector's forward threshold for time checking. Default is 5 minutes (300 seconds).
Past Threshold	The number of seconds by which to extend the connector's rear threshold for time checking. Default is 1 hour (3600 seconds).
Frequency	The SmartConnector checks its future and past thresholds at intervals specified by this number of seconds. Default is 1 minute (60 seconds).
Cache	Changing these settings does not affect the events cached, it only affects new events sent to the cache.
Cache Size	SmartConnectors use a compressed disk cache to hold large volumes of events when the ArcSight Manager is down or when the SmartConnector receives bursts of events. This parameter specifies the disk space to use. The default is 1 GB which, depending on the connector, can hold about 15 million events, but it also can go down to 200 MB. When this disk space is full, the SmartConnector drops the oldest events to free up disk cache space. Select from the options available in the drop-down list. Default is 1 GB.
Notification Threshold	The number of events in the cache that triggers a notification. Default is 10,000 events.
Notification Frequency	How often to send notifications when the notification threshold is reached. Select from the options available in the drop-down list. Default is 10 min.

Configurable Settings, continued

Name Field	Value Field
Maximun File Count	The value set in the user properties represents the maximum number of cache files that guarantees no events dropping, and not the actual amount of cache files created for ingestion. Basically, cache enters in Event drop mode after the number of cache files reaches the limit set. Alternatively, when the number of cache files reaches double the amount set, caching enters into File drop mode.
Network	
Heartbeat Frequency	<p>This setting controls how often the connector sends a heartbeat message to the ArcSight Manager. The default is 5 seconds, but it can go from 5 seconds to 10 minutes. Note that the heartbeat is also used to communicate with the SmartConnector; therefore, if its frequency is set to 10 minutes, then it could take as much as 10 minutes to send any configuration information or commands back to the SmartConnector. Select from the options available in the drop-down list.</p> <p>Default is 10 seconds.</p>
Enable Name Resolution	<p>(No Source/Dest only Yes) The SmartConnector tries to resolve IP addresses to host names, and host names to IP addresses, if the event rate allows it and if required. This setting controls this functionality. The Source, Target and Device IP addresses and Hostnames may also be affected by this setting. The Source/Dest Only choice means that the device address and device host name fields are ignored for name resolution.</p> <p>Default is Yes.</p>
IPv6 Name Resolution Control	<ul style="list-style-type: none"> • IPv4 Only for Legacy Events (default) • IPv6 (Prefer IPv4 for reverse resolution) for Legacy Events • IPv6 (Prefer IPv6 for reverse resolution) for Legacy Events
Name Resolution TTL (secs)	This is the amount of time (Time to Live) the name resolution is to be in effect. The name resolution entries are cached for this time (default is 3600).
Wait For Name Resolution	<p>(Yes No) If set to Yes, the SmartConnector waits for name resolution to be completed. When Yes is selected, event processing might be slowed down significantly and even cause lost events.</p> <p>Default is No.</p>
Name Resolution Host Name Options	<ul style="list-style-type: none"> • Set host name only (default) • Set host name only (lowercase) • Set host and domain names • Set host and domain names (lowercase) <p>For reverse resolution (IP Address to Host name), only the host name field is set. If host name only is not used, the host name is split up and put into both the DNS domain and the host name fields. This affects the source, destination, device and agent address. If one of the (lowercase) choices is made, then the name is changed to lowercase before it is put into the host name (and possibly DNS domain) field(s).</p>
Name Resolution Domain from Email	<p>(Yes No) If set to Yes, the host name and DNS domain fields are empty, and the corresponding user name field appears as an e-mail address, then the domain from the e-mail address is put in the DNS domain field. This only affects the source and destination fields.</p> <p>Default is Yes.</p>

Configurable Settings, continued

Name Field	Value Field
Clear Host Names Same as IP Address	(Yes No) If set to Yes and the host name field is set to an IP Address that matches the corresponding IP Address field, then the host name field is cleared. This affects the source, destination, and device fields. Default is Yes.
Set Host Names to IP Addresses When Unknown	(Yes No) If set to Yes, host names that remain unresolved are set to IP addresses. Default is No.
Don't Resolve Host Names Matching	By default, host names are resolved to their IP addresses. You have the option to specify a regular expression for all or part of a host name <i>for which you do not want the system to attempt host name resolution to an IP address</i> . When this option is configured, the system cannot resolve host names matching this expression.
Don't Reverse-Resolve IP Ranges	By default, IP addresses are resolved to their domain names. You have the option to specify IP address ranges <i>for which you do not want the system to attempt reverse-resolution to domain names</i> . Click in the field to enter the IP address range. To enter a single IP address, enter the address under the From column and leave the To column blank, then click Apply . For an address range, enter the starting IP address under From and the ending address under To , then click Apply . This field lets you to enter a list of ranges. When this option is configured, the system cannot reverse-resolve IP addresses that fall within any of the specified ranges.
Remove Unresolvable Names/IPs from Cache	(Yes Yes (w/ negative cache) No) If set to No , unresolvable host names or IP addresses continue to be in the cache. If set to Yes, unresolvable host names or IP addresses are removed from the cache. If set to Yes (w/negative cache), the connector remembers what names/IPs have been unresolvable so that time is not wasted trying to resolve them frequently. Default is No.
Limit Bandwidth To	Select from a list of bandwidth options you can use to constrain the connector's output over the network. Select from the options available in the drop-down list. Default is Disabled.
Transport Mode	(Normal Cache Cache but send Very High severity events). You can configure the SmartConnector to cache to disk all the processed events it receives. This is equivalent to pausing the SmartConnector. However, you can use this setting to delay event-sending during particular time periods. For example, you could use this setting to cache events during the day and send them at night. You can also set the connector to cache all events, except for those marked with a very-high severity, during business hours, and send the rest at night. Default is Normal.
Cache Mode	(Normal Drop if Dest Down) This option is meant to be used on a primary destination to control the caching behavior of the primary destination when it is down, and the connector starts sending events to the failover destination. In the Normal mode, events are cached and sent to the primary destination when it comes back up. In the Drop if Dest Down mode, the events are not cached and dropped and therefore not sent to the primary destination when it becomes available again. Default is Normal.

Configurable Settings, continued

Name Field	Value Field
Address-Based Zone Population Defaults Enabled	<p>(Yes No) If Yes, the default zones built into the connector will be used to assign zones. These zones are only used if a network model has not been sent by ESM or ArcMC, or if that network model does not cover some addresses. If the Address-Based Zone Population setting (below) is specified, you may want to change this to No.</p> <p>Default is Yes.</p>
Address-Based Zone Population	<p>If specified in setup or ArcMC, this is a comma-separated list that must contain a multiple of three items. The first of each three is the starting IP address of a zone, the second is the ending IP address of the zone, and the third is the URI of the zone to assign to addresses in that range. These zones are only used if a network model has not been sent by ESM or ArcMC, or if that network model does not cover some addresses. If Address-Based Zone Population Defaults Enabled is set to Yes, the zones specified here take precedence over those.</p> <p>For example for two zones this could be: 15.0.0.0,15.255.255.255,/All Zones/ArcSight System/Public Address Space Zones/Hewlett-Packard Company,17.0.0.0,17.255.255.255,/All Zones/ArcSight System/Public Address Space Zones/Apple Computer Inc.</p>
Zone Population Mode	<p>(Normal Rezone (override) No Zoning (clear)) Setting to Normal means zones are computed and assigned, if not already set. Rezone (override) re-computes and re-assigns already populated zones. No Zoning (clear) clears the zones, if already populated.</p> <p>Default is Normal.</p>
Customer URI	<p>Applies the given customer URI to events emanating from the connector. Provided the customer resource exists, all customer fields are populated on the ArcSight Manager. If this particular connector is reporting data that might apply to more than one customer, you can use Velocity templates in this field to conditionally identify those customers.</p>
Field Based Aggregation	<p>Field-based aggregation implements a flexible aggregation mechanism; two events are aggregated if only the selected fields are the same for both events.</p> <p>Note: Field-based aggregation creates a new alert that contains only the fields that were specified, so the rest of the fields are ignored, unless "Preserve Common Fields" is set to "Yes".</p> <p>SmartConnector aggregation significantly reduces the amount of data received, and should be applied only when you use less than the total amount of information the event offers. For example, you could enable field-based aggregation to aggregate "accepts" and "rejects" in a firewall, but you should use it only if you are interested in the count of these events, instead of all the information provided by the firewall.</p>
Time Interval	<p>Choose a time interval, if applicable, to use as a basis for aggregating the events the connector collects. Aggregation time interval and threshold settings need to both be set in order for the aggregation to be enabled. Select from the options available in the drop-down list.</p> <p>Default is Disabled.</p>
Event Threshold	<p>Choose a number of events, if applicable, to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 900 events were found to be the same within the time interval selected (for example, contained the same selected fields) and you select an event threshold of 500, you then receive two events, one of count 500 and another of count 400. This option is exclusive of Time Interval. Select from the options available in the drop-down list.</p> <p>Default is Disabled.</p>

Configurable Settings, continued

Name Field	Value Field
Field Names	Choose one or more fields, if applicable to use as the basis for aggregating the events the connector collects. The result is a comma-separated list of fields to monitor.
Fields to Sum	Choose one or more fields, if applicable, to use as the basis for aggregating the events the connector collects. If specified, this set of numeric fields is summed rather than aggregated, preserved, or discarded. The most common fields to sum are bytesIn and bytesOut . Note that if any of the fields listed here are also in the list of field names to aggregate, they are aggregated and not summed.
Preserve Common Fields	(Yes No) Choosing Yes adds fields to the aggregated event if they have the same values for each event. Choosing No , the default, ignores non-aggregated fields in aggregated events.
Filter Aggregation	Filter Aggregation is a way of capturing aggregated event data from events that would otherwise be discarded due to an agent filter. Only events that would be filtered out are considered for filter aggregation (unlike Field-based aggregation, which looks at all events).
Time Interval	Choose a time interval, if applicable, to use as a basis for aggregating the events the connector collects. It is exclusive of Event Threshold. Select from the options available in the drop-down list. Default is Disabled.
Event Threshold	Choose a number of events, if applicable, to use as a basis for aggregating the events the connector collects. This is the maximum count of events that can be aggregated; for example, if 900 events were found to be the same within the time interval selected (for example, contained the same selected fields) and you select an event threshold of 500, you then receive two events, one of count 100 and another of count 400. This option is exclusive of Time Interval. Select from the options available in the drop-down list. Default is Disabled.
Fields to Sum	(Optional) Choose one or more fields, if applicable, to use as the basis for aggregating the events the connector collects.
Processing	
Preserve Raw Event	<p>(Yes No) Some devices contain a raw event that can be captured as part of the generated alert. If that is not the case, most connectors can also produce a serialized version of the data stream that was parsed/processed to generate the ArcSight event. This feature allows the connector to preserve this serialized "raw event" as a field in the event inspector. This feature is disabled, by default, since using raw data increases the event size and therefore requires more database storage space.</p> <p>You can enable this by changing the Preserve Raw Event setting. If you choose Yes, the serialized representation of the "Raw Event" is sent to the selected destination and preserved in the Raw Event field.</p> <p>Default is No.</p> <div> <p>Note: When selecting Aggregate Events, the Preserve Raw Event feature is disabled.</p> </div>

Configurable Settings, continued

Name Field	Value Field
Turbo Mode	<p>If your configuration, reporting, and analytic usage permits, you can greatly accelerate the transfer of a sensor's event information through SmartConnectors by choosing one of two "turbo" (narrower data bandwidth) modes.</p> <p>Complete is the default transfer mode, which passes all the data arriving from the device, including any additional data (custom, or vendor-specific). This corresponds to turbo.enabled=false on the Manager. Since this value is not the default, be sure to add this property to the Manager's <ARCSIGHT_HOME>/config/server.properties file. After making changes to this file, you need to restart the Manager.</p> <p>The first level of Turbo acceleration is called Faster and drops just additional data, while retaining all other information. The Fastest mode eliminates all but a core set of event attributes, in order to achieve the best throughout. Consider the possible effects such a restricted data set might have from a given device (for example, on reports, rules, threat resolution) before selecting it.</p> <p>The specific event attributes that apply to these modes in your enterprise are defined in the <ARCSIGHT_HOME>/config/server.default.properties file for the ArcSight Manager. Because these properties may have been adjusted in the corresponding server.properties file for your needs, you can refer to this server.properties file for definitive lists. Refer to the ESM Administrator's Guide, topic on "Managing and Changing Properties File Settings" for details.</p> <p>Only scanner SmartConnectors must run in Complete mode, to capture the additional data.</p> <p>Note: SmartConnector Turbo Modes are superseded by the Turbo Mode in use by the ArcSight Managers processing their events. For example, a Manager set to Faster cannot pass all the data possible for a SmartConnector that is set for the default of Complete.</p>
Enable Aggregation (in secs)	<p>Note: If you have already used this feature for setting up previous SmartConnectors, you can continue to do so. However, ArcSight recommends that you use the new "Field Based Aggregation" on page 83 feature as a more flexible option.</p> <p>Here is the description of the legacy "Enable Aggregation" feature, for those who are still using it:</p> <p>When enabled, Enable Aggregation (in seconds) aggregates two or more events on the basis of the selected time value. (Disabled, 1, 2, 3, 4, 5, 10, 30, 60)</p> <p>Default is Disabled.</p> <p>The aggregated event shows the event count (how many events were aggregated into the displayed event) and event type. The rest of the fields in the aggregated event take the values of the first event in the set of aggregated events.</p>
Limit Event Processing Rate	<p>You can moderate the SmartConnector's burden on the CPU by reducing its processing rate. This can also be a means of dealing with the effects of event bursts.</p> <p>The choices range from Disabled (no limitation on CPU demand) to 1 eps (pass just one event per second, making the smallest demand on the CPU).</p> <p>Be sure to note that this option's effect varies with the category of SmartConnector in use, as described in the SmartConnector Processing Categories table.</p>
Fields to Obfuscate	<p>Using MD5 hashing, this option lets you to specify a list of fields for obfuscation in a security event. In FIPS mode, SHA-256 is used.</p>

Configurable Settings, continued

Name Field	Value Field
Store Original Time In	<p>(Disabled Flex Date 1) This parameter lets you to move the original device receipt time to a specified field if altered by the time correction.</p> <p>Default is Disabled.</p>
Enable Port-Service Mapping	<p>(No Yes) If set to Yes and one of the two fields destination port and application protocol is set, and the other is not, the one that is set is used to set the other. For example, if the destination port is 22 and application protocol is not set, then the application protocol is set to ssh.</p> <p>Default is No.</p>
Uppercase User Names	<p>(Disabled Enabled (orig to ID) Enabled(orig to ID or Flex) Enabled(orig to Add. Data))</p> <p>If set to any of the enabled settings, the two user name fields are automatically changed to uppercase.</p> <p>The original values are saved as follows:</p> <ul style="list-style-type: none"> • Enabled (orig to ID) saves the original values to the sourceUserID and destinationUserID fields, respectively, overwriting any values that may have been there previously. • Enabled (orig to ID or Flex) saves the original values in the same fields if they do not already contain values, or to the flexString1 (source) and flexString2 (destination) fields if the ID fields do contain values. • Enabled (orig to Add. Data) saves the original values to additional data fields called OrigSrcUserName and OrigDstUserName, respectively. <p>Note: The uppercase operation is typically done using the default Locale for the chosen platform. You can set this to a particular Locale by setting the connector.uppercase.user.name.locale property in agent.properties to the desired Locale (using "en_US" for U.S. English, for example).</p> <p>Default is Disabled.</p>
Enable User Name Splitting	<p>(Yes No) If this is set to yes and the destination user name contains commas in the event, this parameter duplicates that event. Each user name in the list is placed in one of the events.</p> <p>For example, if the destination user name in an event is "User 123, User 456", then that event is sent twice, with the destination user name set to "User 123" in the first and "User 456" in the second.</p> <p>Default is No.</p>
Split File Name into Path and Name	<p>(Yes No) If this is set to yes and an event's file name field is set but its file path field is not, this parameter splits the file name into a path and a name, placing each part into appropriate fields.</p> <p>For example, if the file name field is set to C:\dir\file.ext and the file path is not set, then the file path is set to C:\dir and the file name to file.ext. The separator character can be either \ or / as the system looks to the SmartConnector to determine its platform.</p> <p>Default is No.</p>

Configurable Settings, continued

Name Field	Value Field
Event Integrity Algorithm	<p>(Disabled SHA-256 SHA-1 MD5 SHA-512)</p> <p>If this is set to one of the algorithms (such as SHA-256), and the Preserve Raw Event parameter is Enabled, then additional event integrity internal events are generated, normally at a rate of about 1 per 50 normal events.</p> <p>The crypto signature field is also set in each event in the format: "#seq(alg):digest", where seq is a persistent event sequence number, alg is the message digest algorithm, and digest is the hexadecimal message digest.</p> <p>These extra events and the crypto signature field values can be used to verify that no events were tampered with after generation.</p> <p>Supported algorithms are: SHA-256, SHA-1, MD5, and SHA-512.</p> <p>Default is Disabled (that is, no algorithm is applied).</p>
Generate Unparsed Events	<p>(Yes No) If set to yes and some incoming event data cannot be parsed (perhaps because a device has been upgraded since the SmartConnector parser was written), then a special event named "Unparsed Event" is generated. The raw event appears in the event message field.</p> <p>If set to No, the SmartConnector log files indicate the unparsed events.</p> <p>Default is No.</p>
Preserve System Health Events	<p>(Yes No) If set to yes, internal system health events are preserved.</p> <p>SmartConnectors generate system health events that provide information about the systems on which they are installed (for example, disk usage, network memory, JVM memory, percentage of processing of CPU memory usage, and so forth). By default, these events are not retained or passed on to ArcSight destinations and, therefore, not available for viewing. Setting this option to yes makes them available in the Console or any destination like Logger.</p> <p>Default is No.</p>

Configurable Settings, continued

Name Field	Value Field
Enable Device Status Monitoring (in millisec)	<p>(<<i>NumberOfMilliseconds</i>> -1 (disabled))</p> <p>If set to a <<i>NumberOfMilliseconds</i>>, the selected SmartConnector generates internal events periodically 1 minute (60000 milliseconds) or greater with the status of the devices for which the connector is receiving normal events. These events have the name "Connector Device Status."</p> <p>Enabling periodic device status monitoring events helps monitor both the SmartConnector and device uptime.</p> <p>Device status monitoring events include this information, if available:</p> <ul style="list-style-type: none"> • Event name (Connector Device Status) • Vendor and Product information • Source Address and Host Name • Zone • Last event received • Total number of events for the device since the connector started • Event count since last call <p>Device status monitoring events can be set to generate every 1 minute (60000 milliseconds), or less frequently (that is, a greater number of milliseconds than the minimum).</p> <p>If you specify less than 60000, you get a warning in the log that the minimum is 60000 milliseconds (1 minute) and the system uses the minimum.</p> <p>If you enter a non-number in the field, it generates an error in the log that the value could not be parsed. In this case, the feature is disabled (and logged as such).</p> <p>In such cases, there is no indication on the Console that anything went wrong because there is no way for the Connector to convey that error.</p>
Payload Sampling (when available)	<p>Some SmartConnectors use Payload sampling to send a portion of packet payload (as opposed to the complete payload) along with the original event. This portion is retrieved using the on-demand payload retrieval in the event inspector.</p>
Maximum Length	<p>You can configure the maximum length of the payload sample using the following values:</p> <ul style="list-style-type: none"> • Discard • 128 bytes • 256 bytes • 512 bytes • 1 Kbyte <p>When the Discard option is chosen, no payload sample is sent inside the original event.</p> <p>Default is 256 bytes.</p>
Mask Non-printable Characters	<p>(False True) This feature lets you to mask the non-printable characters in the payload sample.</p> <p>Default is False.</p>

Configurable Settings, continued

Name Field	Value Field
Filters	Agent severity is the translation of the device severity into normalized values. For example, some connectors use a device severity scale of 1-10, whereas others use a scale of high, medium and low. These values are normalized into a single agent severity scale. The default scale is Low, Medium, High, and Very High. An event can also be classified as Unknown if the data source did not provide a severity rating.
Filter Out	Filters for SmartConnectors are exclusive (filter out). Events that meet the connector filtering criteria are not forwarded to the destination. During SmartConnector set up, you can configure the connector to use filter conditions that do not pass events to the destination according to specific criteria. For example, you can use filters to exclude events with certain characteristics or events from specific network devices.
Very High Severity Event Definition	Enter a filter condition to sort for very high severity events.
High Severity Event Definition	Enter a filter condition to sort for high severity events.
Medium Severity Event Definition	Enter a filter condition to sort for medium severity events.
Low Severity Event Definition	Enter a filter condition to sort for low severity events.
Unknown Severity Event Definition	Enter a filter condition to sort for unknown severity events.

Managing SmartConnector Filter Conditions

Filter conditions to focus the events passed to the destination according to specific criteria can be added during SmartConnector installation and configuration. For example, you can use filters to sort out events with certain characteristics, from specific network devices, or generated by vulnerability scanners. Events that do not meet the connector filtering criteria are not forwarded.

See “Managing SmartConnector Filter Conditions” in the “Managing SmartConnectors” chapter of the *ArcSight ESM Console User Guide* for filters that can be applied through the ESM Console. This guide can be found on [Protect 724](#) under ArcSight Product Documentation for ArcSight ESM and ESM Express. A filter applied through the ESM Console only applies to the events sent to that ESM.

For all other types of destinations, the filter must be expressed in text, as described below. For many connectors you can specify filter conditions to narrow the scope of the events to be processed. For example, you can write filtering strings such as:

Name EQ “Agent”

(name Contains “Super”) Or (name EQ “Agent”)

attackerAddress Between (“10.0.0.1”, “10.0.0.10”)

destinationAddress Is “NOT NULL”

The following table lists usable operators. For more information about data fields, event mappings, and CEF fields, see the “Data Fields,” “Audit Events,” “Cases,” and “Events” sections in *ArcSight ESM User’s Reference*.

Usable Operations	Description
EQ	equal to
NE	not equal to
LT	less than
LE	less than or equal to
GE	greater than or equal to
GT	greater than
Between	compares any specified range
ContainsBits	equal to, for bitmap fields
In	standard CCE operator for membership test
Contains	contains the specified string
StartsWith	starts with the specified string
EndsWith	ends with the specified string
Like	standard CCE operator for simple pattern matching for string type: _ wildcard for single character, % wildcard for any number of characters
InSubnet	for IP address that is not the specified subnet
InGroup	for asset in the specified asset category or zone in the specified zone group
Is	tests true for the selected state, “NULL” or “NOT NULL” . Do not use all uppercase of “Is”.

Chapter 9: ArcSight Logger SmartMessage (encrypted) Destination

ArcSight Logger is a log management solution that is optimized for extremely high event throughput. Logger logs (or stores) time-stamped text messages, called events, at high sustained input rates. Events consist of a receipt time, a source (host name or IP address), and an un-parsed message portion. Logger compresses raw data, but also can retrieve it in an unmodified form for forensics-quality litigation reporting. Unlike ESM, Logger does not normalize events.

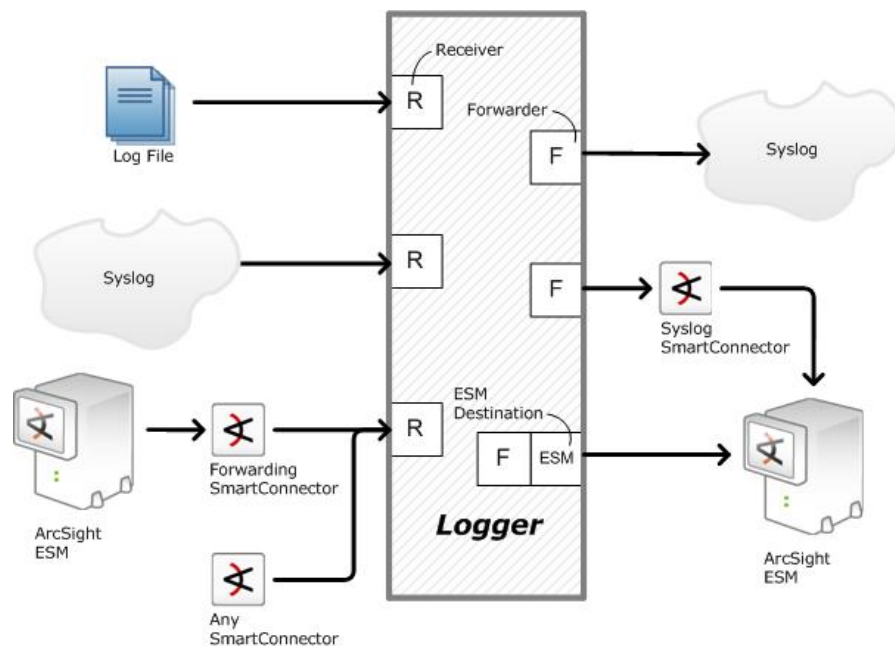
Multiple Loggers can work together to support an extremely high event volume. Logger can be configured as a peer network with queries distributed across all peer Loggers.

Sending Events from Logger to a Manager

Logger's most basic function is to store a large volume of security events. Logger can send a subset of these events to a Manager. It sends syslog or ArcSight Common Event Format (CEF) events directly to ESM through a built-in connector called an ESM Destination. An ESM Destination appears as a connector on a Console. For more information about ESM Destinations, see the *ArcSight Logger Administrator's Guide*.

SmartMessage is ArcSight technology used by Logger to provide a secure channel between connectors and Logger. SmartMessage provides an end-to-end encrypted secure channel. At one end is an connector, receiving events from the many devices it supports; on the other end is SmartMessage Receiver on Logger.

Note: To forward events from Logger to ESM, use a Syslog connector to send events to Logger. If a different method such as Netcat is used, the events are forwarded to Logger but not to ESM.

Logger Receivers (R) and Forwarders (F)

Note: The SmartMessage secure channel uses HTTPS (secure sockets layer protocol) to send encrypted events to Logger. This is similar to, but different from, the encrypted binary protocol used between connectors and the ESM Manager.

Use port 443 (rather than ArcSight traditional port 8443) because the secure channel uses HTTPS.

Sending Events to Logger

1. Set up the SmartMessage Receiver on Logger (see the *ArcSight Logger Administrator's Guide* for detailed instructions).
2. Install the connector component as documented in the configuration guide for the connector being installed.
3. Navigate through the windows and select **ArcSight Logger SmartMessage (encrypted)**. See ["Connector Destinations" on page 72](#) to view the options.
4. Click **Next**. Enter the **Logger Host Name/IP**, leave the port number at default (443) or change it to 9000 if the destination is a software logger, and enter the **Receiver Name**. This setting should match

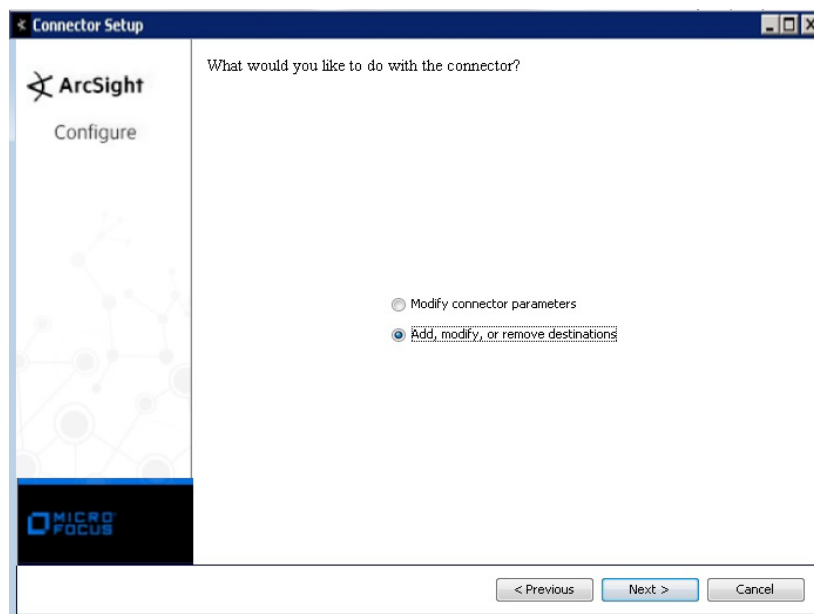
the Receiver name you created in step 1 so that Logger can listen to events from this connector.

Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name.
Compression Mode	The data compression mode checkbox. Select to enable or leave as default for disable.
CEF Version	<p>Select 0.1 or 1.0 from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes.</p> <p>0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to 2³¹-1).</p> <p>1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to 2⁶³-1).</p>

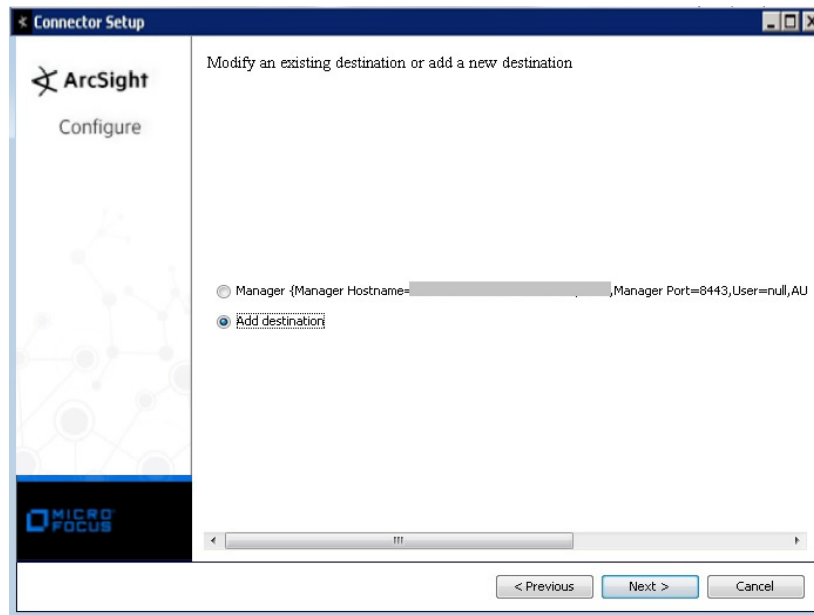
1. Click **Next**. If you haven't already imported the certificate, the Logger certificate message to import the certificate to connector displays.
2. Ensure the **Import the certificate to connector from destination** option is selected and click **Next**.
3. Navigate through the subsequent windows until receiving a message that confirms the configuration was successful. Select **Exit** and click **Next** to exit the wizard.

Sending Events to Both Logger and a Manager

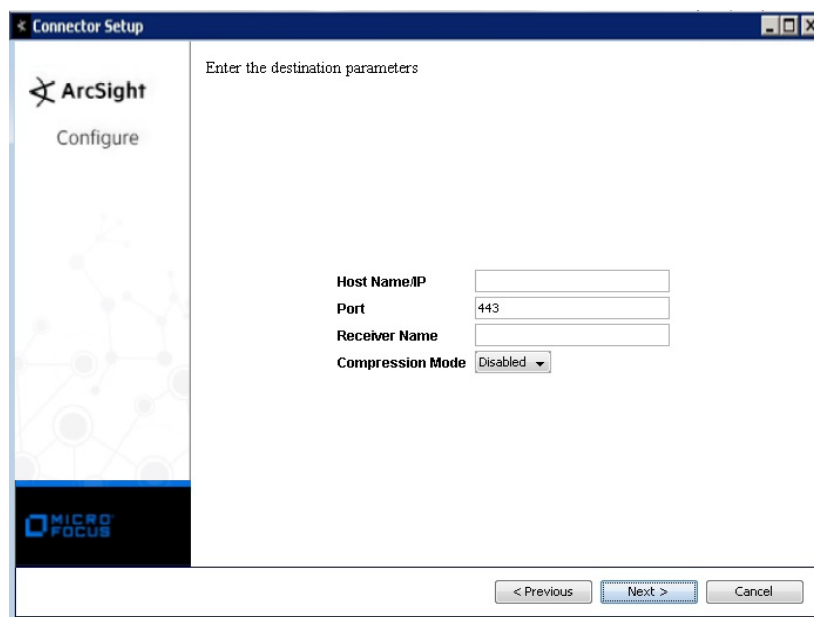
1. Set up the SmartMessage Receiver on Logger (see the configuration guide for the connector being installed).
2. Install the connector component (see the connector Configuration Guide for your device).
3. Register the connector with a running ESM Manager and test that the connector is up and running.
4. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
5. Select **Add, modify, or remove destinations**.



6. Click **Next**. Select **Add destination**.



7. Click **Next**. Select **ArcSight Logger SmartMessage (encrypted)**. See ["Connector Destinations" on page 72](#) to view the options.
8. Click **Next**. Enter the **Logger Host Name/IP**, leave the port number at default (443) or change it to 9000 if the destination is a software logger, and enter the **Receiver Name**.



9. Click **Next**. If you haven't already imported the certificate, the Logger certificate message to import the certificate to connector displays.
10. Ensure the **Import the certificate to connector from destination** option is selected and click **Next**.
11. Click **Next**. A message confirms that the configuration was successful. Select **Exit** and click **Next** to exit the wizard.

- Restart the connector for changes to take effect.

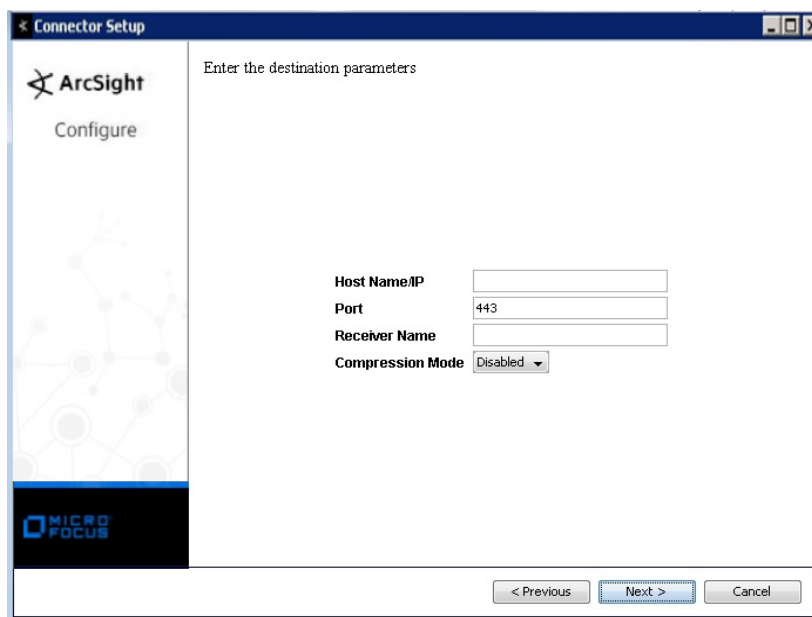
Forwarding Events from ESM to Logger

The ArcSight Forwarding Connector can read events from an ESM Manager and forward them to Logger using ArcSight's Common Event Format (CEF).

Note: The Forwarding Connector is a separate installable file, named similarly to this: **ArcSight-6.x.x.<build>.x-SuperConnector-<platform>.exe**.

Use Forwarding Connector build 4810 or later for compatibility with Logger 1.5 or later.

- Follow the instructions in the connector Configuration Guide for your device to install the connector.
- When you see the type of destination window, select **ArcSight Logger SmartMessage (encrypted)**. See ["Connector Destinations" on page 72](#) to view the options.
- Click **Next**. Enter the **Logger Host Name/IP**, leave the port number at default (443) or change it to 9000 if the destination is a software logger, and enter the **Receiver Name**.



The screenshot shows the 'Connector Setup' window with the 'Configure' tab selected. The title bar says 'Connector Setup'. The main area is titled 'Enter the destination parameters'. On the left, there is an ArcSight logo and the word 'Configure'. Below that is a Micro Focus logo. The main area contains the following fields:

- Host Name/IP: [Text box]
- Port: 443 [Text box]
- Receiver Name: [Text box]
- Compression Mode: Disabled [Dropdown menu]

At the bottom, there are three buttons: '< Previous', 'Next >', and 'Cancel'.

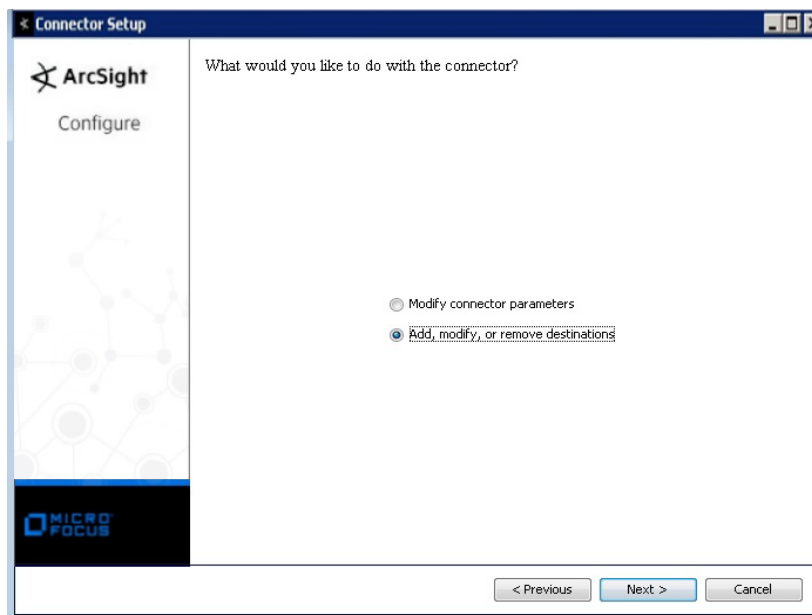
- Click **Next**. You will get a Logger certificate message to import the certificate to connector.
- Ensure the **Import the certificate to connector from destination** option is selected and click **Next**.
- Click **Next**. A message confirms that the configuration was successful. Select **Exit** and click **Next** to exit the wizard.
- Restart the connector for changes to take effect.

To configure the Forwarding Connector to send CEF output to Logger and send events to another Manager at the same time, see ["Sending Events to Both Logger and a Manager" on page 94](#).

Defining Connector Settings in Logger

After installing the connectors to communicate with Logger, you can set up their properties through the connector Configuration Wizard. Assuming you have installed the connector component as previously shown (see ["Installing Connectors" on page 30](#) for detailed instructions), complete these steps:

1. Using the `$ARCSIGHT_HOME\current\bin\runagentsetup` script, restart the connector configuration program.
2. Select **Add, modify, or remove destinations.**



3. Click **Next**. For details, see ["Configuring Connectors" on page 46](#).
4. Click **Next** and proceed with the configuration.

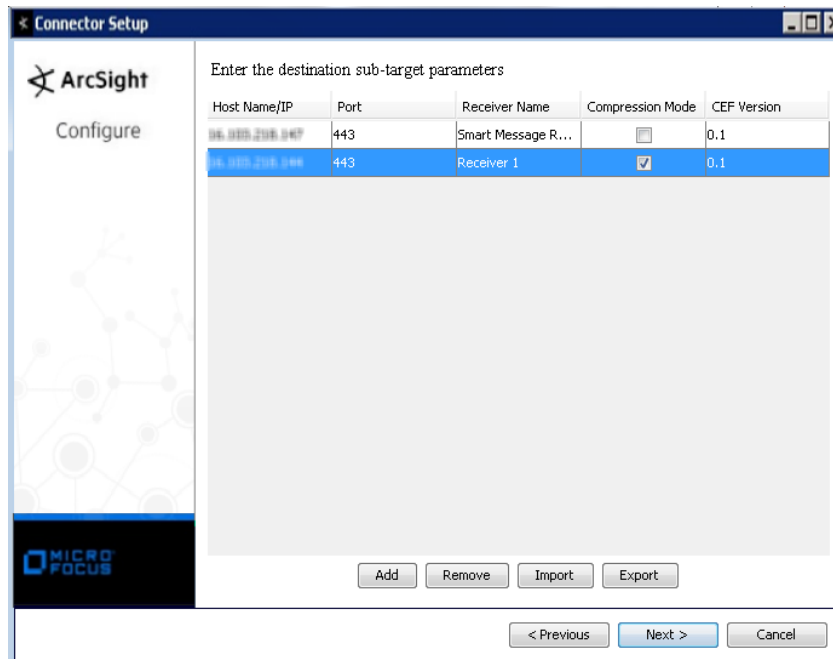
Chapter 10: ArcSight Logger SmartMessage Pool (Encrypted) Destination

Use the ArcSight SmartMessage Logger Pool (encrypted) destination type to specify a pool of logger devices. The pool is a single destination consisting of one or more loggers. Each “Batch” of events processed by the connector is sent to the next logger in the pool in a round-robin fashion. A batch is typically 100 events although you can configure the batch size. If a pool member is unavailable, events are sent to the remaining pool members. After a pool member becomes available again, the connector resumes sending events to it. If no pool members are available, the events are sent to the failover destination.

Note: When Logger SmartMessage Pool destination is used, the connector cannot be managed through the Management Center 2.0 and earlier versions.

Configuring a Logger Pool Destination

1. Set up the SmartMessage Receiver on all Loggers that you plan to include in the LoggerSecure Pool (see the *ArcSight Logger Administrator's Guide* for detailed instructions).
2. Install the connector component as documented in the configuration guide for the connector being installed.
3. Navigate through the windows until you see the destination types window. Select **ArcSight Logger SmartMessage Pool (encrypted)**. See ["Connector Destinations" on page 72](#) to view the options.
4. Click **Next** to continue and add the logger pool members.

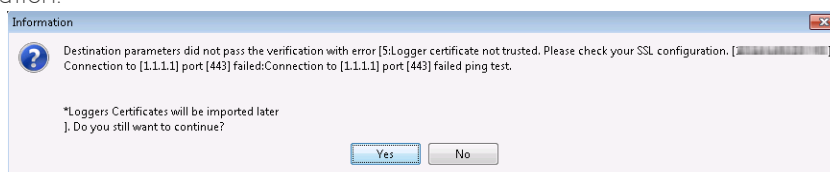


- Click **Add** to add each logger pool member and enter the host name, port number, and receiver name fields. The parameters and buttons are described in the following tables.

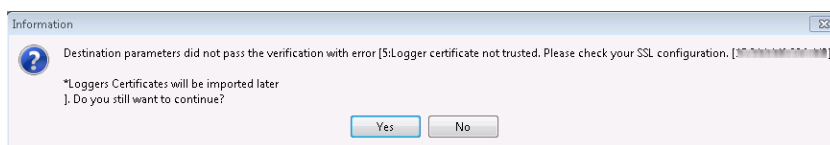
Parameter	Description
Host Name/IP	The destination host name or IP address.
Port	The destination port 443 for Logger Appliance or 9000 for Software Logger.
Receiver Name	The destination receiver name.
Compression Mode	The data compression mode checkbox. Select to enable or leave as default for disable.
CEF Version	<p>Select 0.1 or 1.0 from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes.</p> <p>0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to 2³¹-1).</p> <p>1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to 2⁶³-1).</p>

Button	Description
Add	Adds a row to the table to add a logger to a pool. Fill in the information manually. Use the checkbox for Compression Mode to enable or disable it. The default is unchecked for disabled. The default port for logger is 443.
Remove	Removes the row corresponding to the logger from the loggersecure pool.
Import	Opens a dialog window to import the .csv file type containing the pre-recorded information for loggersecure pool.
Export	Opens a dialog window where you can export and save the data entered in the panel. Use a .csv file extension for export. The file lists Disabled for default Compression Mode and TRUE for enabled.

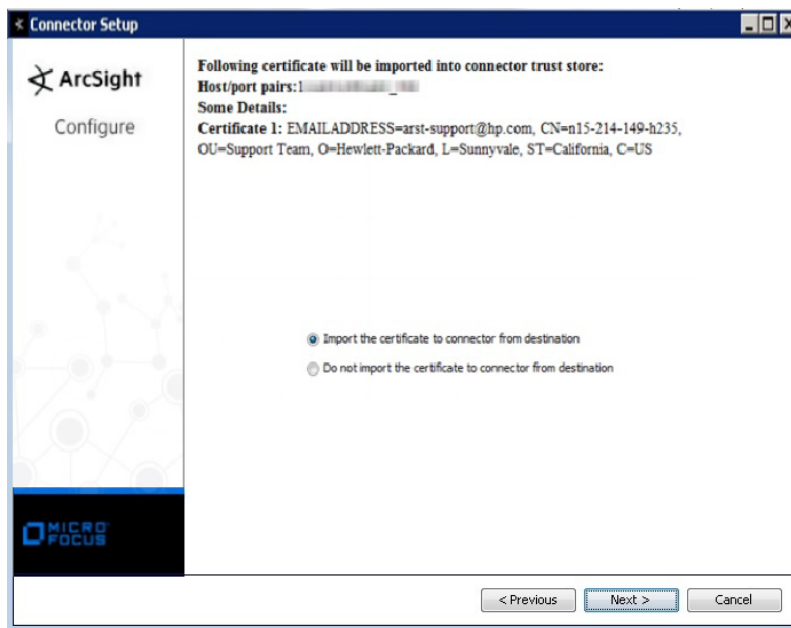
6. Continue until all pool members are added, then click **Next**.
 - a. If any of the parameters could not pass the verification, an error displays asking you to check your SSL configuration.



- b. Check the connectivity error by clicking **No** and return to the parameter window to edit the parameter for the logger that has the error.
 - c. Click **Next** again to continue with configuration.
 - d. You will see a message asking you if you want to continue. Click **Yes**.



- e. Enter the name of the Connector location and click **Next**.
7. You will get a Logger certificate message to import the certificate to connector. Ensure the **Import the certificate to connector from destination** option is selected and click **Next**.



Persisting SmartMessage Transport

Occasionally, depending on the network, the connector could experience problems sending a batch of events to Logger. During that time, the following symptoms might be noticed in the log: Logger ping test could fail frequently; the EPS could drop down; the heartbeat transport and event transport links could sporadically go up and down. In the statistics, longer roundtrip times might be observed for 'event sent' acknowledgment, events could fail to be sent, and caching may be observed.

You can make SmartMessage transport persistent to achieve higher throughput for Logger destinations by modifying the following property to change the value to **true** in the **agent.properties** file (located at **\$ARCSIGHT_HOME\current\user\agent**):

transport.loggersecure.connection.persistent=true

Changing the persistent value to **true** is not recommended if there are more than 250 Logger connections.

Chapter 11: CEF Destinations

This chapter explains the following selections available for sending events in Common Event Format (CEF). Note that the Transformation Hub destination can send events in either CEF format or binary format.

- [CEF File](#)
- [Transformation Hub](#)
- [CEF Syslog](#)
- [CEF Encrypted Syslog \(UDP\)](#)

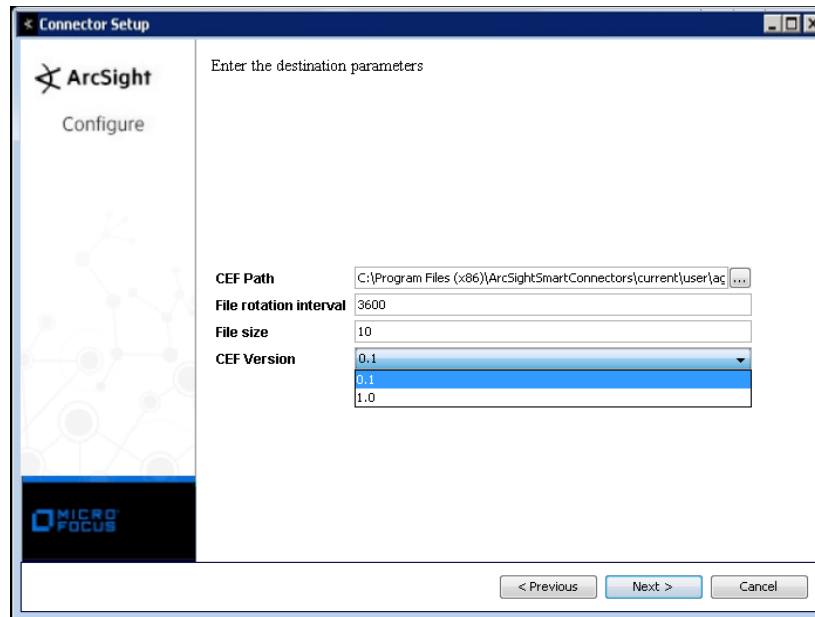
CEF File

This selection lets you capture events that a connector would normally send to the ESM Manager, and route them to a file. The format called Common Event Format (CEF) can be readily adopted by vendors of both security and non-security devices. This format contains the most relevant event information, making it easy for event consumers to parse and use them.

For detailed descriptions of field information, see the *Cloud CEF Implementation Standard*.

1. To proceed, run the Installation Wizard and choose **CEF File**. See ["Connector Destinations" on page 72](#) to view the options.
2. Enter the following values for these parameters.

Parameter	What to enter or select
CEF Folder	Path where the CEF files are stored
File Rotation Interval	The desired file rotation interval, in seconds. The default is 3,600 (one hour).
File Size	File size in megabytes (default: 10 MB)
CEF Version	<p>Select 0.1 or 1.0 from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination could be Logger, another SmartConnector, or a non-ArcSight product.</p> <p>0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to $2^{31}-1$).</p> <p>1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to $2^{63}-1$).</p>



3. Click **Next** and proceed with the installation.

File Rotation

Events are appended to the current file until the rotation time interval expires or the maximum file size is reached. When either condition is exceeded, a new current file is created and the previous current file is renamed (as detailed below).

Event files are named using the timestamp of their creation, and all files, with the exception of the current file, have the text '**done.cef**' appended. For example, a typical CEF file set configured to rotate every hour might consist of files named in this manner:

2010-01-28-10-55-33.cef

2010-01-28-09-55-33.done.cef

2010-01-28-08-55-33.done.cef

Transformation Hub

The Transformation Hub destination is used to send events to an Transformation Hub cluster, which can then further distribute events to real-time analysis and data warehousing systems. Any application that supports retrieving data from Transformation Hub can receive these events (for example, ESM, ArcSight Investigate, Hadoop and Logger).

Note: The configuration settings for ESM must be done on the connector side, not the ESM Console.

Specify the event topic name. All connectors that use the same logger pool need to be configured to use the same event topic name, so the events from these connectors will be published to the same event topic.

For Content Types CEF 0.1 and CEF 1.0, the key is sent on events with the connectors IP address and a flag. The flag format is a single byte value. For ESM, the key is the agent ID.

The key format is: one byte flags + (4 or 16 bytes) IP (v 4 or v 6) address. Based on the value of the IP version bit, 4 or 16 additional bytes should be examined. This is used in case the key is made longer in a non-breaking fashion in the future.

Bit position	Meaning
0	IP version: 0 = IPv4 1 = IPv6
1	Key version: Must be 0. If there are future versions of key that are not backward compatible with this definition, it changes to 1.
2-7	Key version: Must be 0. Reserved for future.

For CEF 0.1 and 1.0, the events are delivered to Transformation Hub in their own messages, which are distributed to the partitions of the topic defined in Transformation Hub in a round-robin manner. For ESM, the events are sent in batches in a binary format. TLS encryption is supported, as is client certificate authentication.

When TLS is enabled by setting the **Use SSL/TLS** parameter to **true** during destination configuration, a Java KeyStore-format (.jks) file containing the certificates of the Transformation Hub's Kafka cluster, or a certificate that has signed them, will be required. The location of this Trust Store file will be required during destination configuration. See Kafka documentation at https://kafka.apache.org/documentation.html#security_ssl for instructions.

Also, when client certification authentication is enabled by setting the **Use SSL/TLS Authentication** parameter to **true**, a .jks file containing the private key and certificate to use must be provided. The Transformation Hub cluster must have the certificate (or a certificate that has signed it) in its trust store. The location of the Key Store file and authentication information is to be provided in the **SSL/TLS Key Store file**, **SSL/TLS Key Store pass**, and **SSL/TLS Key password** parameters. The Key and Key Store passwords are created when you set up Transformation Hub.

1. To proceed, run the Installation Wizard and choose **Transformation Hub** as the destination.
2. Enter values for the following parameters.

Parameter	What to enter or select								
Initial Host:Port(s)	<p>This is a required field. Provide a comma-separated list of hostnames and ports for establishing communication with the Transformation Hub cluster. Not all servers in the cluster must be listed, but if none of the servers listed can be contacted, the connector cannot send events to Transformation Hub. Specify at least one server. An example would be:</p> <p><code>kafka1.example.com:9093,kafka2.example.com:9093.</code></p>								
Content Type	<p>Select these Topics for the corresponding Content Types:</p> <table> <tr> <th>Content Type</th><th>Topic</th></tr> <tr> <td>Logger/Investigate/Hadoop/3rd parties</td><td> eb-cef Supports IPv4. Use with Logger 6.3.0 or earlier versions. </td></tr> <tr> <td>Logger 6.4 or higher/IPv6/Investigate</td><td> eb-cef Supports IPv4 and IPv6. Use with Logger 6.4.0 or later versions. In addition to IPv6 support, this option adds support for long values for Bytes In/Out fields. </td></tr> <tr> <td>ESM</td><td> eb-esm See the section on "ESM Support of Other ArcSight Products/Components" in the ArcSight ESM Support Matrix. </td></tr> </table>	Content Type	Topic	Logger/Investigate/Hadoop/3rd parties	eb-cef Supports IPv4. Use with Logger 6.3.0 or earlier versions.	Logger 6.4 or higher/IPv6/Investigate	eb-cef Supports IPv4 and IPv6. Use with Logger 6.4.0 or later versions. In addition to IPv6 support, this option adds support for long values for Bytes In/Out fields.	ESM	eb-esm See the section on "ESM Support of Other ArcSight Products/Components" in the ArcSight ESM Support Matrix .
Content Type	Topic								
Logger/Investigate/Hadoop/3rd parties	eb-cef Supports IPv4. Use with Logger 6.3.0 or earlier versions.								
Logger 6.4 or higher/IPv6/Investigate	eb-cef Supports IPv4 and IPv6. Use with Logger 6.4.0 or later versions. In addition to IPv6 support, this option adds support for long values for Bytes In/Out fields.								
ESM	eb-esm See the section on "ESM Support of Other ArcSight Products/Components" in the ArcSight ESM Support Matrix .								
Acknowledgment mode	<p>This is a required field. The value selected determines whether the connector is to wait for acknowledgment from Transformation Hub that it has received the event. Options are:</p> <p>Leader: Default. The connector waits for acknowledgment from the primary Transformation Hub server for the event's partition. This option protects against data loss in most circumstances while providing reasonable performance; however, throughput can be affected.</p> <p>None: The connector does not wait for acknowledgment. This can result in lost events if the receiving Kafka server fails, but has significantly higher throughput.</p> <p>All: The connector waits for an acknowledgment from all Transformation Hub servers that contain a backup for the event's partition. This protects against lost events in nearly all circumstances, but significantly reduces throughput.</p>								
Use SSL/TLS	<p>Determines whether events are sent with TLS encryption. Options are:</p> <ul style="list-style-type: none"> True False (default) <p>If true is selected, the SSL/TLS Trust Store Password and the location of the SSL/TLS Trust Store file are required.</p>								
SSL/TLS Trust Store file	Enter the location of the Trust Store file.								
SSL/TLS Trust Store password	Enter the password for the SSL/TLS Trust Store.								

Parameter	What to enter or select
Use SSL/TLS Authentication	<p>Determines whether a client certificate is used for TLS to identify the connector. Options are:</p> <ul style="list-style-type: none"> • True • False (default) <p>When true is selected, Use SSL/TLS must also be enabled. The values for the SSL/TLS Key Store File, SSL/TLS Key Store Pass, and SSL/TLS Key password parameters also must be provided.</p>
SSL/TLS Key Store file	Enter the location of the SSL/TLS Key Store file.
SSL/TLS Key Store pass	Enter the password for SSL/TLS Key Store.
SSL/TLS Key password	Enter the password for SSL/TLS Key.

The screenshot shows the 'Connector Setup' window for ArcSight. The title bar reads 'Connector Setup@n... com'. The window has a sidebar with the ArcSight logo and the word 'Configure'. The main area is titled 'Enter these parameters' and contains the following fields and dropdowns:

- Initial Host:Port(s): [Empty text box]
- Content type: [Logger/Investigate/Hadoop/3rd parties dropdown]
- Topic (hover for recommendations): [eb-cef text box]
- For ESM topic, the ESM version: [6.11.x dropdown]
- Acknowledgment mode: [leader dropdown]
- Use SSL/TLS: [false dropdown]
- SSL/TLS Trust Store file: [Empty text box]
- SSL/TLS Trust Store password: [Empty text box]
- Use SSL/TLS Authentication: [false dropdown]
- SSL/TLS Key Store file: [Empty text box]
- SSL/TLS Key Store pass: [Empty text box]
- SSL/TLS Key password: [Empty text box]

At the bottom of the window, there are three buttons: '< Previous', 'Next >', and 'Cancel'.

3. Click **Next** and proceed with the installation.

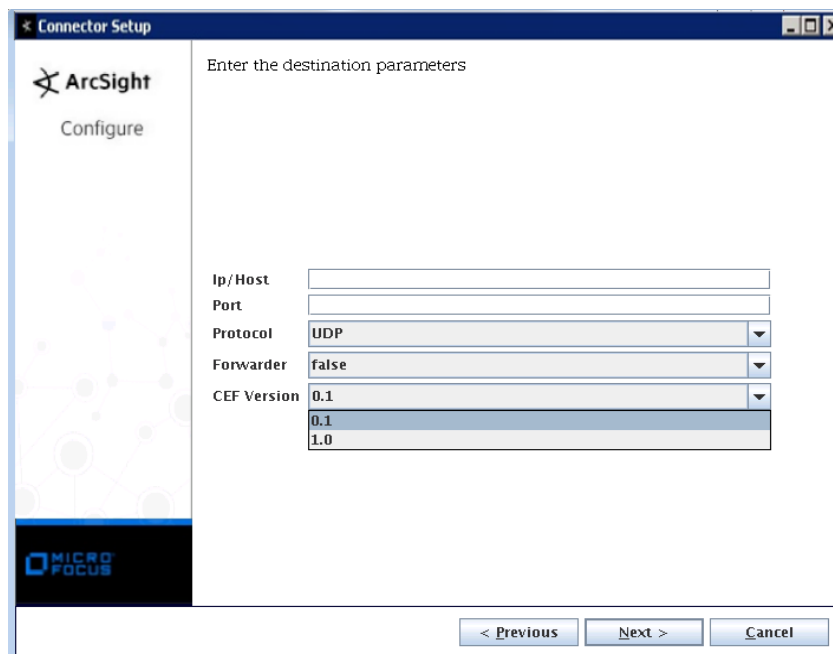
CEF Syslog

The **TCP** and **UCP** destination can be used to send events to Logger, where data is received using a TCP or UDP Receiver. One such receiver can receive from more than one connector. These can also be used to send to a Syslog Daemon connector or non-ArcSight syslog receivers.

For detailed information about sending to Logger, see ["ArcSight Logger SmartMessage \(encrypted\) Destination" on page 91](#).

The **TLS** protocol provides a means of sending events through a secure channel (an option that does not apply to Logger). This data can be received by any application that supports TLS syslog reception, which includes ArcSight's Syslog NG Daemon connector.

1. Install the connector following the instructions in the configuration guide for your device. You may also see the following window after you add a destination, see ["Add, Modify, or Remove Destinations"](#) on [page 47](#).
2. When you see the type of destination window, choose **CEF Syslog**. See ["Connector Destinations"](#) on [page 72](#) to view the options.
3. Click **Next**.
4. Enter the following values for these parameters.



Parameter	What to enter or select
IP/Host	Enter the IP/ Host information.
Port	Enter the Port information.

Parameter	What to enter or select
Protocol	Select the appropriate protocol from the drop-down menu.
Forwarder	<p>The CEF Forwarder mode parameter is False by default. If the destination is a Syslog Daemon connector and you want to preserve information about the original connector, then the CEF Forwarder mode should be set to True both in this destination and in the receiving connector. In other words, if you have a chain of connectors connected by syslog, syslog NG, or CEF encrypted syslog (UDP), and you want to preserve information about the original connector, the destinations should all have the CEF Forwarder mode set to True (which is implicitly true for CEF Encrypted Syslog (UDP)), and the connectors receiving from them should also have the CEF Forwarder mode set to True.</p> <p>For example, you can configure a number of forwarders for Microsoft Windows Event Log Unified, all sending events using the CEF Syslog destination type to one Syslog Daemon connector, which then sends to ESM. In order for the events arriving at ESM to retain information about the specific Unified connector that collected the event, the connector's CEF Syslog destinations should have the Forwarder mode set to true, and the Syslog Daemon connector should also set the Forwarder mode to true. The information will display in the original agent fields of the events.</p>
CEF Version	<p>Select 0.1 or 1.0 from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination could be Logger, another SmartConnector, or a non-ArcSight product.</p> <p>0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to $2^{31}-1$).</p> <p>1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to $2^{63}-1$).</p>

- Click **Next** to proceed with the remainder of the installation.

Reconnect Feature for Load Balancing

If you have a multiple tier connector installation where there is a load balancer between tiers, you can use the reconnect feature for better load balancing behavior. For example, without the reconnect feature, tier 1 connectors start up and make a connection to the CEF syslog destination (tier 1). The load balancer makes a load balancing decision at the time of the initial connection and the tier 1 connector always sends to that same tier 2 connector.

With the **reconnect** parameter, the tier 1 connector makes an initial connection to the tier 2 connector as before and the load balancer makes a load balancing decision and picks a tier 2 connector. But, after the reconnect timeout, the tier 1 connector makes a new connection and the load balancer makes a new load balancing decision and selects a tier 2 connector that could be a different tier 2 connector connected previously. This spreads the load evenly across the tier 2 connectors over time.

To make use of the reconnect parameter:

1. From `$ARCSIGHT_HOME/current/user/agent`, open the `agent.properties` file for editing.
2. Locate the following parameter to edit:
`agents[0].destination[0].params`
3. Change the value for reconnect from "-1" to the number of seconds the CEF syslog destination should stay open before a disconnect and reconnect is performed.

For example, change:

```
<Parameter Name=\"reconnect\" Value=\"-1\"/>\n
```

to

```
<Parameter Name=\"reconnect\" Value=\"60\"/>\n
```

This enables the disconnect and reconnect to be performed every minute.

4. Save and exit `agent.properties`.

CEF Encrypted Syslog (UDP)

The CEF Encrypted Syslog (UDP) destination allows for events to be sent encrypted over UDP, using a "Shared Secret".

Caution: Logger does not accept CEF Encrypted Syslog.

To decrypt the data on the receiving side, ensure that you have installed and configured the ArcSight CEF Encrypted Syslog (UDP) connector. If the connector is not yet installed, refer to the *SmartConnector for ArcSight CEF Encrypted Syslog (UDP)* for instructions.

1. Install the connector component (see the connector Configuration Guide for your device). You may also see the following window after you add a destination, see ["Add, Modify, or Remove Destinations" on page 47](#).
2. When you see the type of destination window, choose **CEF Encrypted Syslog (UDP)**. See ["Connector Destinations" on page 72](#) to view the options.
3. Click **Next**.
4. Enter the following values for these parameters.

Parameter	What to enter or select
IP/Host	Enter the IP/Host.
Port	Enter the Port information.
CEF Version	<p>Select 0.1 or 1.0 from the drop-down menu. Select 0.1 if you are not sure the chosen destination can handle CEF 1.0, which supports both IPv4 and IPv6 modes. The destination can only be the corresponding SmartConnector.</p> <p>0.1 - The Device Address, Source Address, Destination Address, and Agent Address fields will always be IPv4 or will be omitted. If there are any IPv6 addresses, they will be in Device Custom IPv6 Address fields. Bytes In and Bytes Out fields are limited to the size of an integer (up to 2³¹-1).</p> <p>1.0 - Any of the address fields can be either IPv4 or IPv6 and the Bytes In and Bytes Out fields can be long values (up to 2⁶³-1).</p>
Shared Key (16 characters)	Enter a 16 character shared key for encryption (Shared Secret). The same Shared Key must be used when configuring the CEF Encrypted Syslog (UDP) connector on the receiving side.

- Click **Next** and proceed with the installation.

Chapter 12: CSV File Destination

This chapter explains how to capture events that a connector would normally send to the ESM Manager, and route them to a file. Typical ArcSight configurations do not require the use of external files to communicate events to the ESM Manager.

Event data is written to a file in Excel-compatible comma-separated values (CSV) format, with comments prefixed by '#'. A connector can be configured to preface the data with a comment line that describes the fields found on a subsequent line. An example of a typical event file:

```
#event.eventName,event.attackerAddress,event.targetAddress
```

```
"Port scan detected","1.1.1.1","2.2.2.2"
```

```
"Worm ""Code red"" detected","1.1.1.1","2.2.2.2"
```

```
"SQL Slammer detected","1.1.1.1","2.2.2.2"
```

```
"Email virus detected","1.1.1.1","2.2.2.2"
```

Event data is written to files in the specified folder and can be configured to rotate periodically.

CSV File Installation

1. To install a connector that logs security events in a CSV file rather than forwarding them to an ESM Manager, run the connector Installation Wizard and, from the destination selection window, choose **CSV File**. See ["Connector Destinations" on page 72](#) to view the options.
2. Enter or select values for these parameters.

The screenshot shows the 'Connector Setup' window for ArcSight. The title bar says 'Connector Setup'. On the left, there's an ArcSight logo and the word 'Configure'. The main area is titled 'Enter the destination parameters'. It contains four fields: 'CSV Path' (empty), 'Fields' (containing 'event.deviceReceiptTime,event.r'), 'File rotation interval' (containing '3600'), and 'Write format header' (a dropdown menu set to 'false'). At the bottom right, there are three buttons: '< Previous', 'Next >', and 'Cancel'.

Parameter	What to enter or select
CSV Path	The path to the output folder. If it does not exist, the folder is created.
Fields	<p>A comma-delimited list of field names to be sent to the CSV file. The default is: event.deviceReceiptTime,event.name,event.deviceAddress, event.deviceHostName,event.sourceAddress, event.sourceHostName,event.sourcePort, event.destinationAddress,event.destinationHostName, event.destinationPort</p> <p>To modify the list, each entry needs to begin with either:</p> <ul style="list-style-type: none"> • “event.” and the name of a normal pre-defined event field, or • “additionaldata.” and the name of some additional data field that applies to this particular connector. These names are not common across all connectors. <p>There are no spaces allowed around the commas in the field names. For example: “event.deviceReceiptTime,event.name” is correct. But, “event.deviceReceiptTime, event.name” is not correct.</p>
File rotation interval	Enter the desired file rotation interval, in seconds. The default is 3,600 (one hour).
Write format header	Select true to send a header row with labels for each column, as described above.

- Click **Next** and proceed with the installation.

Event Data Rotation

Events are appended to the current file until the rotation time interval expires, at which time a new current file is created and the previous current file is renamed. One hour is a typical rotation time interval.

Event files are named using the timestamp of their creation, and all files, with the exception of the current file, have the text '**done.csv**' appended. For example, a typical CSV file set configured to rotate every hour might consist of files named in this manner:

2007-01-28-10-55-33.csv

2007-01-28-09-55-33.csv.done

2007-01-28-08-55-33.csv.done

Using the properties file, the configuration of your CSV connector can be customized to filter and aggregate events as desired.

A connector can also be configured to send events to a CSV file and an ESM Manager at the same time.

Chapter 13: Raw Syslog Destination

This chapter explains how to capture raw syslog events. See the *Connector Configuration Guide for Raw Syslog Daemon* for information about both the connector and the Raw Syslog destination.

Raw Syslog Overview

Although normalized data is faster and easier to parse and access, many IT professionals prefer having the raw data available for review, forensics, and litigation. In conjunction with the Raw Syslog connector destination, the connector for Raw Syslog Daemon lets you extract and collect raw syslog events from syslog servers using the TLS, Raw TCP, or UDP protocols.

Note: If you are transporting data to ArcSight Logger, you can use specific configuration parameters to provide minimal normalization of the syslog data (for source and timestamp). See the *SmartConnector for Raw Syslog Daemon Configuration Guide* for details.

This destination works in conjunction with the Raw Syslog connector, which captures security events in raw syslog. When you install the Raw Syslog connector, run the connector Installation Wizard and, from the destination selection, choose **Raw Syslog**. See ["Connector Destinations" on page 72](#) to view the options.

After you enter the raw syslog destination parameters and click **Next**, the connector Configuration Wizard proceeds through the configuration process.

Appendix A: ArcSight Update Packs (AUPs)

This appendix describes the ArcSight Update Packs (AUPs) used to update content between the ESM Manager and connectors. AUP files may contain information that applies to connectors or ESM related updates.

ArcSight Content AUPs

AUP files provide a way to collect a set of files together and update ArcSight resources as well as distribute parsers to connectors. ArcSight continuously develops new connector event categorization mappings, often called "content." This content is packaged in ArcSight Update Packs (AUP) files. All existing content is included with major product releases, but it is possible to stay completely current by receiving up-to-date, regular content updates from Micro Focus Subscribers Choice. Contact Micro Focus SSO for details.

Content updates (**ArcSight-xxxx-ConnectorContent.aup**) are provided by support and contain data that is then transferred to registered connectors. An AUP can provide updates for:

1. Event categorizations (Category Behavior, Category Object, etc.)
2. Default zone mappings (what IP maps to which zone by default)
3. OS mappings (when a network is scanned, where the asset is created)

Content such as filters, rules and dashboards is not provided by the AUP.

Note: Management Center do not support automatic deployment of an AUP.

As shown below, the method of uploading an AUP varies depending on the ArcSight product.

ESM

Content updates are available from support. To update,

1. Download the latest AUP release.
2. Copy the **.aup** file to **ARCSIGHT_HOME\updates** onto a running ESM Manager. Connectors registered to this ESM automatically download the **.aup** and, once completed, an audit event is generated.

ESM/Logger

A connector can send events to ESM and Logger simultaneously. In this configuration, it's helpful to use the AUP Master Destination feature. AUP Master Destination allows ESM to push AUP content to the connector

used for its Logger destination(s). Logger is not capable of storing or pushing its own AUP content.

1. Using the connector Configuration Wizard, add the ESM destination and set the AUP Master Destination parameter to **true** (the default is false).
2. If you have not already done so, you can also add the Logger destination.
3. Copy the **.aup** file to **ARCSIGHT_HOME\updates** on the running ESM Manager you added in step 1.

Connector

The AUP content is pushed from ESM to the connector, which then sends an internal event to confirm. If the AUP Master Destination flag was set for the ESM destination, that AUP content is used by the connector for Logger or any other non-ESM destinations.

Caution: The AUP Master Destination flag should be set to **true** for only one ESM destination at a time. If more than one ESM destination is set and the flag is true for more than one, only the first is treated as master.

Failover ESM destinations cannot be AUP Masters.

Logger

Logger has no facility to store or forward AUPs to connectors.

Management Center

To use AUP content through Management Center, use the AUP/ENC repository. This tool lets you maintain a number of connector AUP (upgrade) files. You can apply any of these AUP upgrade files to container to upgrade to a specific version. As a result, all connectors in a container are upgraded to the version you apply to the container.

For instructions on how to upgrade, refer to “Upgrade AUP/ENC Repository” in the *Micro Focus ArcSight Management Center Administrator’s Guide*.

ESM Generated AUPs

Some AUPs are generated by ESM itself for internal maintenance and operation.

User Categorization Updates

User Categorization Updates, (for example, **user-categorizations_user_supplied_00000000001300014581.aup**) are generated by ESM when a user modifies the way an event is categorized through the Console tools. These updates are then transferred to the registered connectors to update the way the newly sent events will be categorized. This is generally used for categorizing custom signatures for which ArcSight does not provide categorization.

System Zones Updates

System Zones updates (for example, **system-zone-mappings_00000000000000000001.aup**) are generated by ESM when a change to the ArcSight System zones is detected, then transported to the necessary connectors. It contains the new System-Zone mappings so incoming events are attached to the correct zones or assets in ESM.

As System Zones are always present, all connectors connected to ESM routinely receive them as an AUP.

User Zones Updates

User Zones updates (for example, **user-zone-mappings_3Rxkk0xYBABDRZ1Zyr6nrWg==_00000000001700001895.aup**) are generated by ESM when a change to a user-created zone configuration is detected, then transported to the necessary connector. It contains the new zone mappings so that incoming events are attached to the correct zones or assets in ESM.

Appendix B: FIPS Compliant SmartConnectors

This appendix describes the FIPS configuration and installation.

What is FIPS?

Under the Information Technology Management Reform Act (Public Law 104-106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for Federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use government-wide. NIST develops FIPS when there are compelling Federal government requirements such as for security and interoperability and there are no acceptable industry standards or solutions.

FIPS Suite B includes cryptographic algorithms for hashing, digital signatures, and key exchange. The entire suite of cryptographic algorithms is intended to protect both classified and unclassified national security systems and information.

Note: When FIPS-compliant connectors connect to a non-FIPS-compliant destination, the solution is not considered FIPS compliant. Also, when the destination is installed in FIPS Suite B compliant mode, the SmartConnectors also must be installed in FIPS Suite B compliant mode.

Which Connectors are Supported?

FIPS Compliant Connectors

- All syslog connectors
- All file reader connectors
- All SNMP connectors
- Most database connectors (except Oracle Audit DB and when using SQL Server drivers with encryption)
- Cisco Secure IPS SDEE connectors
- Sourcefire Defense Center eStreamer connector
- Check Point OPSEC NG connector

FIPS Non-Compliant Connectors

- Database connectors using SQL Server drivers with encryption
- Connectors using Oracle drivers
- Connectors running on AIX or Micro Focus UX platforms only

Connectors Not Certified as FIPS Compliant

- Various API connectors with proprietary internal mechanisms
- Web Services and Cloud connectors

Connector Caveats

Certain limitations apply for some connector types, as described in the sections that follow.

CEF Syslog as the Destination

If you choose **CEF Syslog** (with TLS protocol) as the destination for the connector, the wizard attempts to retrieve the security certificate from the destination and import it based upon your input. Although the CEF Syslog destination works as expected in FIPS-compliant mode, when you edit **agent.properties** to enable FIPS-compliant mode (as described in ["Enable FIPS Support"](#)), the certificate retrieved from the destination may not be imported properly into the truststore.

If the SmartConnector wizard is unable to fetch and import the destination certificate, you can import the certificate manually:

1. Copy the certificate from the destination to a temporary location.
2. From the **\$ARCSIGHT_HOME/current/bin** directory, execute the following command to import the certificate:
arcsight keytoolgui
3. Open the keystore in **\$ARCSIGHT_HOME/jre/lib/security/cacerts** (the password will be **changeit**).
4. From the **Menu** bar, select **Tools** and **Import Certificate**. Upload the certificate file.
5. Trust the certificate.
6. Start the connector and the device.

Microsoft SQL JDBC Driver

If you are running a database connector that uses the SQL JDBC driver *with encryption enabled*, the connector cannot be installed in FIPS-compliant mode.

See the configuration guide for the database connector you are installing for instructions for downloading and installing a Microsoft SQL Server JDBC driver.

Enable FIPS Support

When installing a software connector, the instructions for enabling FIPS Support is provided as part of the installation procedure. During installation and configuration of the connector, on the "Set Global Parameters" window, set to **Enable** to enable FIPS-Compliant mode. To enable FIPS Suite B mode through the wizard, see ["Enabling FIPS Suite B Mode"](#).

When installing a SmartConnector on an appliance, you can enable FIPS support through the user interface. To do this, enable support on the container or containers containing the connector for which you want to enable support.

Manually Enable FIPS Mode

1. From `$ARCSIGHT_HOME/current/user/agent`, open the `agent.properties` file for editing.
2. Enter the following property:
`fips.enabled=true`
3. Save and exit `agent.properties`.

Manually Enable FIPS Suite B Support

If you have installed a SmartConnector in FIPS-compliant mode, you can manually enable FIPS Suite B support by modifying the ESM destination parameters in the `agent.properties` file as follows:

Note: The destination must also be installed in FIPS Suite B mode.

1. From `$ARCSIGHT_HOME\current\user\agent`, open the `agent.properties` file for editing.
2. Locate the following property for destination parameters (approximately, line 10 in the file):
`agents[0].destination[0].params=<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<ParameterValues>\n <Parameter Name=\"port\" Value=\"8443\"/>\n <Parameter Name=\"filterevents\" Value=\"false\"/>\n <Parameter Name=\"host\" Value=\"samplehost.sv.arcsight.com\"/>\n <Parameter Name=\"aupmaster\"`


```
Value\="false"/>\n <Parameter Name\="fipsciphers"  
Value\="fipsDefault"/>\n</ParameterValues>\n
```

3. The destination parameters are specified here as an XML string where each element is one parameter. Based upon the Suite B mode of the destination, change **fipsDefault** to **suiteb128** (for 128-bit security) or **suiteb192** (for 192-bit security).
4. Save and exit **agent.properties**.

Restart the connector for your changes to take effect.

Password Management

Use the commands below to change your key and trust store passwords. Then update the **agent.properties** file with the new value.

To change password on a key/trust store:

1. Run the following command (see table below for store value):
bin/arcsight agent keytool -store <store value> -storepasswd
2. Enter the new password as prompted.
3. Update **agent.properties**, according to the table below.

Note Key store files will not exist unless client authentication has been setup.

To change password of a key inside the key store:

A key entry uses the same password as the key store, so when changing the key store password, also change the key's password.

```
bin/arcsight agent keytool -store agentkeys -keypasswd -alias <alias of key>
```

Store Values

Key Store (for Client Authorization)	Trust Store
agentkeys	agentcerts

Entries for **agent.properties** File

When changing passwords, make sure to add or update the corresponding property value in **agent.properties**.

	Key Store (for Client Authorization)	Trust Store
FIPS	ssl.fips.keystore.password= <new password>	ssl.fips.truststore.password= <new password>
Non-FIPS	ssl.keystore.password= <new password>	ssl.truststore.password= <new password>

Appendix C: Add Destinations

Connectors send a copy of events to each additional destination for which it is configured. Additional destinations can be useful, for example, when you have a development ArcSight environment working in parallel with your production environment and you want to test rules and reports. You can configure multiple destinations and also have failover destinations for when the primary destination is unavailable.

In such cases, you can configure the connector to send alerts to both your production Manager and your development Manager to be able to view real-time event flows on both systems. Because the destinations are independent, you do not compromise the events sent to the production Manager. For more information on how to add, refer to ["Add, Modify, or Remove Destinations" on page 47](#).

Client Authentication

Follow these instructions to enable Client Authentication for Transformation Hub and SmartConnectors with Transformation Hub destinations. For additional Transformation Hub configuration, see the **Administrator's Guide** for Transformation Hub and "Transformation Hub" in the **Smart Connector User Guide** on the [Micro Focus Community](#).

Note: The following examples:

- Use the default password. See the appendix for FIPS Compliant SmartConnectors in the **SmartConnector User Guide** on the [Micro Focus Community](#) to set a non-default password.
- Assume that you are on the Linux platform. For Windows platforms, use backslashes when entering commands.
- Assume you are using a command prompt window to enter Windows commands. Do not use Windows PowerShell.

Diagnosing Common Transformation Hub Issues

The following can help to diagnose common Transformation Hub issues.

502 Gateway Error (On Multi-Master Install)

If the initial master node is disabled and later then re-enabled, the user may see a 502 Gateway error when attempting to log in to the Installer UI, preventing login. If this happens, restart all of the **nginx-ingress-controller** pods in the cluster as follows:

1. Determine the nginx pod names: **kubectl get pods -n default**
2. For each nginx pod, run: **kubectl delete pod <nginx name>**

Potential DNS Resolution Issue

The Transformation Hub application pods that depend on hostname resolution from DNS could fail. For example, the Schema Registry pod will be in a crash loop status, with following error message in the Schema Registry logs:

```
# kubectl logs eb-schemaregistry-1138097507-1jxbn -n eventbroker
```

```
...
```

```
org.apache.kafka.common.config.ConfigException: No resolvable bootstrap urls  
given in bootstrap.servers
```

```
...
```

The following steps will be useful in debugging this DNS resolution issue. The key is that the bootstrap host name given should be resolvable from within the pod, which can be verified as follows. Find the schema registry pod name:

```
# kubectl get pods -n eventbroker1 | grep schemaregistry
```

```
eb-schemaregistry-2567039683-9l9jx    1/1      Running    1      18d
```

Find the configured bootstrap server

```
# kubectl logs eb-schemaregistry-2567039683-9l9jx -n eventbroker1 | grep  
"bootstrap.servers ="
```

```
bootstrap.servers = [<hostname>]
```

Use **ping** to check if the hostname is DNS resolvable. If it is resolvable, you will see a successful ping.

```
# kubectl exec eb-schemaregistry-2567039683-9l9jx -n eventbroker1 -- ping -  
c 1 n15-214-137-h51.arst.usa.hp.com | grep transmitted
```

```
1 packets transmitted, 1 packets received, 0% packet loss
```

If not, you will see an error

```
# kubectl exec eb-schemaregistry-2567039683-9l9jx -n eventbroker1 -- ping -  
c 1 bad.dns.hostname.arst.usa.hp.com | grep transmitted
```

```
ping: unknown host
```

If hostname is not resolvable, please check the DNS configuration on the system.

Transformation Hub Cluster Down

The number of nodes required to keep an Transformation Hubcluster operating depends on the replication factor. If the replication factor is only 1, which is not recommended, then all Kafka nodes in the EB cluster need to be up to make the TH cluster function correctly. In general, if the replication factor is N, then the system will tolerate up to N-1 server failures without losing any records committed to the log.

Pod Start Order

After deployment, pods are configured to start in the following order (downstream pods will not start until the dependencies are met.)

1. A quorum of ZooKeeper pods in the cluster must be up (2 of 3, or 3 of 5). The total number of ZooKeepers must be odd.
2. All Kafka pods must be up
3. Schema Registry pod must be up
4. Bootstrap Web Service, Transformation Hub Manager
5. Transformation Stream Processor, Routing Stream Processor

Cannot query ZooKeeper

This can occur when running the **kubect1 get pods** command to get status of the pods, downstream pods (as defined in the pod start order) do not stay up, and status is a 'CrashLoop'-type error.

- Ensure ZooKeeper pods are running.
- If the ZooKeeper pod status is **Pending**, you may not have labeled the nodes correctly(zk=yes). Verify that the nodes are labeled using the **kubect1 get nodes -L=zk** command.
- Verify that you configured an odd number of ZooKeepers in the [arcsight-installer.properties](#) eb-zookeeper-count attribute.
- Check the ZooKeeper pod logs for errors using the **kubect1 logs <pod name> -n eventbroker1**.

Common Errors and Warnings in ZooKeeper logs

- **Quorum Exceptions:** A leader cannot be elected. If you see this type of error, check the conditions above.
- **Socket error:** this can occur if there are too many connections. The solution is to restart the pod using the **kubect1 delete pod <pod_name> -n eventbroker1**. The pod will be recreated automatically.

Common Errors and Warnings in Kafka logs

Cannot Register ID: In some cases, a broker node cannot register its ID. This can be caused by multiple broker nodes with the same ID. This is a rare situation that can occur when you are adding and removing nodes from the cluster and you do not define the cluster properly. Connect to each system running a Kafka broker and check the assigned `broker.id` value of each, in `/opt/arcsight/k8s-hostpath-volume/eb/kafka/meta.properties`. The `broker.id` value defined on each Kafka node must be unique.

SSL Connection Errors: These are warnings that occur if there is a connection issue between Kafka and consumer or producer.

Cannot communicate with other brokers: Host names may not be configured properly. It is possible that the node cannot perform reverse lookup or that DNS is not set up properly.

Transformation Hub default topics not created on first deployment: In this instance, the Bootstrap Web Service log contains 500 response code (the response from the Schema Registry), and topics are not created. Try undeploying the Transformation Hub containers, and then redeploy them.

One or more connectors cannot send data to Kafka: Check the following:

- The connection configuration is set properly in the connector.
- The encryption mode (TLS, TLS+FIPS, TLS+CA, TLS+FIPS+CA) is the same for both the Connector and Transformation Hub.
- Make sure you can connect to the Kafka port on the system and that there are no network issues.

Cannot retrieve the certificate error when connecting: Make sure that time is synced across all systems in the data pipeline.

- Check whether the Kafka pod is down. Did you configure the connector with only one broker address and that broker is down; If you expect that there are multiple brokers, they must be all configured in connector as a comma-separated list;
- If the replication factor is set to 1 and a Kafka broker is down, data will not be sent through Transformation Hub. Fix the broker issue to bring it back up. In general, topics should be configured with replication factor greater than 1 so as to prevent this scenario.

Kafka is resyncing: This may cause event throughput slowdown, but will not stop event flow.

An EB component crashes: Check the following:

- Check the container start up order (above). Have any of the dependency pods not started or crashed?
- It could be that the JVMs require more memory than the system has available.
- Check the number of open sockets.

Transformation Hub EPS is lower than expected: Check resource constants on Transformation Hub nodes, such as CPU, memory, or disk space. Also, check usage with ArcMC.

Network bottleneck: In this case, the Stream Processor is not able to keep up with transformation, or is resource-constrained in some way. In Management Center, the Stream Processor metric will be lower than the connector EPS. Check that you have sufficient resources, memory, CPU.

Continuous network failures: This may be related to the management of TCP/IP resources. TIME_WAIT is the parameter which indicates the amount of time the node will try take to finish closing a connection and the amount of time before it will kill a stale connection. Try reducing the value from its default. Edit the file `/etc/sysctl.conf` and add these lines to the end of it (or edit the existing values):

Decrease TIME_WAIT seconds

```
net.ipv4.tcp_fin_timeout = 10
```

Recycle and Reuse TIME_WAIT sockets more quickly

```
net.ipv4.tcp_tw_recycle = 1
```

```
net.ipv4.tcp_tw_reuse = 1
```

After editing the file you should run

```
$ sysctl --system
```

Configure an Transformation Hub Destination Without FIPS Mode

Follow these steps to configure an Transformation Hub destination from the SmartConnector without client authentication in FIPS mode.

On the SmartConnector Server

1. Prepare the SmartConnector:
 - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
 - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.
2. Navigate to the connector's **current** directory, for example:

```
cd <install_dir>/current
```
3. Set the environment variables for the static values used by keytool, for example:

```
export CURRENT=<full path to this "current" folder>
```

```
export BC_OPTS="-storetype BCFKS -providername BCFIPS -providerclass  
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath  
${CURRENT}/lib/agent/fips/bc-fips-1.0.0.jar  
-J-Djava.security.egd=file:/dev/urandom"
```

```
export EB="<eb hostname>_<eb port>"
```

```
export STORES=${CURRENT}/user/agent/stores
```

4. Create the **user/agent/stores** directory if does not already exist, for example:

```
mkdir ${STORES}
```

5. Copy the **ca.crt** certificate from the Transformation Hub to the connector's **\${STORES}** directory.

The certificate will typically be in the **/opt/kubernetes/ssl** directory on Transformation Hub.

6. Import Transformation Hub certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.crt -alias  
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass <trust store  
password>
```

When prompted, enter **yes** to trust the certificate.

7. Note the trust store path

```
echo ${STORES}/${EB}.truststore.bcfips
```

8. Navigate to the **bin** directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

On Windows platforms: Run the **runagentsetup.bat** executable instead.

- a. When completing the Transformation Hub destination fields, use the same values as above for path and password.
 - b. Set **Use SSL/TLS** to **true**.
 - c. Set **Use SSL/TLS Authentication** to **false**.
9. Cleanup. Delete the certificate file, for example:

```
rm ${STORES}/ca.crt
```

Configure an Transformation Hub Destination with FIPS Mode

Follow these steps to configure an Transformation Hub (TH) destination from the SmartConnector with client authentication in FIPS mode.

Step 1: On the Connector Server

1. Prepare the connector:

- **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode** Set to **Enabled**.
- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's **current** directory, for example:

```
cd <install_dir>/current
```

3. Apply the following workaround for a Java keytool issue:

- a. Create a new file, **agent.security**, at **<install_dir>/current/user/agent**

- b. Add the following content to the file and save:

```
security.provider.1=org.bouncycastle.jcajce.provider
.BouncyCastleFipsProvider
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
security.provider.3=sun.security.provider.Sun
```

4. Set the environment variables for static values used by keytool:

```
export CURRENT=<full path to this "current" folder>
export BC_OPTS="-storetype BCFKS -providertype BCFIPS
-J-Djava.security.egd=file:/dev/urandom
-J-Djava.ext.dirs=${CURRENT}/jre/lib/ext:${CURRENT}/lib/agent/fips
-J-Djava.security.properties=${CURRENT}/user/agent/agent.security"
export EB="<event broker hostname>_<event broker port>"
export STORES=${CURRENT}/user/agent/stores
```

5. Create the **user/agent/stores** directory if it does not already exist, for example:

```
mkdir ${STORES}
```

6. Create the Connector key pair, for example:

```
jre/bin/keytool ${BC_OPTS} -genkeypair -alias ${EB} -keystore
${STORES}/${EB}.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=HP,L=Sunnyvale,ST=CA,C=US" -validity 365
```

When prompted, enter the password. Note the password; you will need it again in a later step. Press Enter to use the same password for the key.

7. List the key store entries. There should be one private key.

```
jre/bin/keytool ${BC_OPTS} -list -keystore ${STORES}/${EB}.keystore.bcfips
-storepass <key store password>
```

8. Create a Certificate Signing Request (CSR), for example:

```
jre/bin/keytool ${BC_OPTS} -certreq -alias ${EB} -keystore  
${STORES}/${EB}.keystore.bcfips -file ${STORES}/${EB}-cert-req -storepass  
<key store password>
```

Step 2: On the Transformation Hub Server

1. Navigate to an empty, temporary directory.
2. Copy the Transformation Hub's root certificate to the following default directory:

```
cp /opt/kubernetes/ssl/ca.crt
```

3. Copy the Transformation Hub's private key to this directory:

```
cp /opt/kubernetes/ssl/ca.key
```

Step 3: On the Connector Server

Copy the **\${STORES}/\${EB}-cert-req** file from the connector to the Transformation Hub directory created above.

Step 4: On the Transformation Hub Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA ca.crt -CAkey ca.key -in <event broker hostname>_  
<event broker port>-cert-req -out <event broker hostname>_<event broker  
port>-cert-signed -days 365 -CAcreateserial -sha256
```

Step 5: On the Connector Server

1. Copy the **\${EB}-cert-signed** certificate from the Transformation Hub to the connector's **\${STORES}** directory.
2. Copy the **ca.crt** certificate from the Transformation Hub to the connector's **\${STORES}** directory.
3. Import the Transformation Hub certificate to trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.crt -alias  
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass <trust store  
password>
```

4. Enter a new password as prompted. Note this as the "trust store" password. You can use the same password as above.

5. Import the Transformation Hub certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.crt -alias  
CARoot -keystore ${STORES}/${EB}.keystore.bcfips -storepass <trust store  
password>
```

6. When prompted, enter **yes** to trust the certificate.

7. Import the signed certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${EB}-cert-signed  
-alias ${EB} -keystore ${STORES}/${EB}.keystore.bcfips -storepass <key  
store password>
```

If successful, this command will return the message, **Certificate reply was installed in keystore.**

8. Note the key store and trust store paths

```
echo ${STORES}/${EB}.truststore.bcfips  
echo ${STORES}/${EB}.keystore.bcfips
```

9. Navigate to the **bin** directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

On Windows platforms: Run the **runagentsetup.bat** executable instead.

- a. When completing the Transformation Hub destination fields, use the same values as above for the path and password.
 - b. Set **Use SSL/TLS** to **true**.
 - c. Set **Use SSL/TLS Authentication** to **true**.
10. Cleanup. Delete the following files:

```
rm ${STORES}/ca.crt  
rm ${STORES}/${EB}-cert-signed  
rm ${STORES}/${EB}-cert-req
```

Step 6: On the Transformation Hub Server

To cleanup the Transformation Hub server, delete the temporary folder where the certificate was signed.

Configure an Transformation Hub Destination with Client Authentication in FIPS Mode

Follow these steps to configure an Transformation Hub (TH) destination from the SmartConnector with client authentication in FIPS mode.

Step 1: On the Connector Server

1. Prepare the connector:

- **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode** Set to **Enabled**.
- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's **current** directory, for example:

```
cd <install_dir>/current
```

3. Apply the following workaround for a Java keytool issue:

- a. Create a new file, **agent.security**, at **<install_dir>/current/user/agent** (or at **<install_dir>\current\user\agent** on Windows platforms).

- b. Add the following content to the file and save:

```
security.provider.1=org.bouncycastle.jcajce.provider
.BouncyCastleFipsProvider
security.provider.2=com.sun.net.ssl.internal.ssl.Provider BCFIPS
security.provider.3=sun.security.provider.Sun
```

- c. Move the **lib/agent/fips/bcprov-jdk14-119.jar** file to the **current** directory.

4. Set the environment variables for static values used by keytool:

```
export CURRENT=<full path to this "current" folder>
export BC_OPTS="-storetype BCFKS -providername BCFIPS
-J-Djava.security.egd=file:/dev/urandom
-J-Djava.ext.dirs=${CURRENT}/jre/lib/ext:${CURRENT}/lib/agent/fips
-J-Djava.security.properties=${CURRENT}/user/agent/agent.security"
export EB=<event broker hostname>_<event broker port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
export EB_HOST={EB master host IP}
export CA_CERT=/tmp/ca.cert.pem
export INTERMEDIATE_CA_CERT=/tmp/intermediate.cert.pem
export FIPS_CA_TMP=/opt/fips_ca_tmp
```

On Windows platforms:

```
set CURRENT=<full path to this "current" folder>
set BC_OPTS=-storetype BCFKS -providername BCFIPS
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security
```

```
set EB=<event broker hostname>_<event broker port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
set EB_HOST={EB master host IP}
set CA_CERT=C:\Temp\ca.cert.pem
set INTERMEDIATE_CA_CERT=C:\Temp\intermediate.cert.pem
set FIPS_CA_TMP=/opt/fips_ca_tmp
```

5. Create the **user/agent/stores** directory if it does not already exist, for example:

```
mkdir ${STORES}
```

On Windows platforms:

```
mkdir %STORES%
```

6. Create the connector key pair, for example (the connector **FQDN**, **OU**, **O**, **L**, **ST**, and **C** values must be changed for your company and location):

```
jre/bin/keytool ${BC_OPTS} -genkeypair -alias ${EB} -keystore
${STORES}/${EB}.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -genkeypair -alias %EB% -keystore
%STORES%\%EB%.keystore.bcfips -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF ,L=Sunnyvale,ST=CA,C=US" -validity 365
```

When prompted, enter the password. Note the password; you will need it again in a later step. Press **Enter** to use the same password for the key. If you want to match the default value in the properties file, use the password **changeit**.

7. List the key store entries. There should be one private key.

```
jre/bin/keytool ${BC_OPTS} -list -keystore ${STORES}/${EB}.keystore.bcfips
-storepass ${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -list -keystore %STORES%\%EB%.keystore.bcfips
-storepass %STORE_PASSWD%
```

8. Create a Certificate Signing Request (CSR), for example:

```
jre/bin/keytool ${BC_OPTS} -certreq -alias ${EB} -keystore
${STORES}/${EB}.keystore.bcfips -file ${STORES}/${EB}-cert-req -storepass
${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -certreq -alias %EB% -keystore  
%STORES%\%EB%.keystore.bcfips -file %STORES%\%EB%-cert-req -storepass  
%STORE_PASSWD%
```

Step 2: On the Transformation Hub Server

1. When EB is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in **/tmp** with the following names:

```
/tmp/intermediate.cert.pem
```

```
/tmp/intermediate.key.pem
```

```
/tmp/ca.cert.pem
```

Use the following command to add them to Transformation Hub:

```
/opt/arcsight/kubernetes/scripts/arcsight-cert-util.sh write --re-  
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-  
ca=/tmp/ca.cert.pem
```

Note: If TH was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy TH, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

2.

```
export CA_CERT=/tmp/ca.cert.pem  
export INTERMEDIATE_CA_CERT=/tmp/intermediate.cert.pem  
export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem  
export FIPS_CA_TMP=/opt/fips_ca_tmp  
export EB=<event broker hostname>_<event broker port>
```
3. Create a temp location on the eb master server:

```
mkdir $FIPS_CA_TMP
```

Step 3: On the Connector Server

Copy the `${STORES}/${EB}-cert-req` file (`%STORES%\%EB%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above, `/opt/fips_ca_tmp`.

Step 4: On the Transformation Hub Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA $INTERMEDIATE_CA_CRT -CAkey $INTERMEDIATE_CA_KEY -  
in ${EB} -cert-req -out ${EB} -cert-signed -days 365 -CAcreateserial -sha256
```

Step 5: On the Connector Server

1. Copy the `${EB}-cert-signed` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the `%EB%-cert-signed` certificate to the connector's `%STORES%` directory.)
2. Copy the `ca.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
3. Copy the intermediate `.cert.pem` certificate from the Transformation Hub to the connector's `${STORES}` directory. (On the Windows platform, copy the certificate to the `%STORES%` directory.)
4. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias  
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias  
CARoot -keystore %STORES%\%EB%.truststore.bcfips -storepass %STORE_  
PASSWD%
```

5. Import the intermediate certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file  
${STORES}/intermediate.cert.pem -alias INTCARoot -keystore  
${STORES}/${EB}.truststore.bcfips -storepass ${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file  
%STORES%\intermediate.cert.pem -aliasINTCARoot -keystore  
%STORES%\%EB%.truststore.bcfips -storepass %STORE_PASSWD%
```

6. Import the CA certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias  
CARoot -keystore ${STORES}/${EB}.keystore.bcfips -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias  
CARoot -keystore %STORES%\%EB%.keystore.bcfips -storepass %STORE_  
PASSWD%
```

7. When prompted, enter **yes** to trust the certificate.
8. Import the intermediate certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file  
${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.keystore.bcfips -storepass ${STORE_  
PASSWD}
```

If successful, this command will return the message, **Certificate reply was installed in keystore**.

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file  
%STORES%\intermediate.cert.pem -alias  
INTCARoot -keystore %STORES%\%EB%.keystore.bcfips -storepass %STORE_  
PASSWD%
```

9. Import the signed certificate to the key store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/${EB}-cert-signed  
-alias ${EB} -keystore ${STORES}/${EB}.keystore.bcfips -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\%EB%-cert-signed  
-alias %EB% -keystore %STORES%\%EB%.keystore.bcfips -storepass %STORE_  
PASSWD%
```

If successful, this command will return the message, *Certificate reply was installed in keystore.*

10. Navigate to the **bin** directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

On Windows platforms:

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.
 - b. Set **Use SSL/TLS** to **true**.
 - c. Set **Use SSL/TLS Authentication** to **true**.
11. Cleanup. Delete the following files:

Caution: The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/intermediate.cert.pem  
rm ${STORES}/intermediate.key.pem  
rm ${STORES}/${EB}-cert-signed  
rm ${STORES}/${EB}-cert-req
```

On Windows platforms:

```
del %STORES%\intermediate.cert.pem
del %STORES%\intermediate.key.pem
del %STORES%\%EB%-cert-signed
del %STORES%\%EB%-cert-req
```

12. Move the `bcprov-jdk14-119.jar` file back to the `lib/agent/fips` directory (or `lib\agent\fips` on Windows platforms).

Step 6: On the Transformation Hub Server

To cleanup the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

Caution: The temporary folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

Configure an Transformation Hub Destination with Client Authentication in Non-FIPS Mode

Follow these steps to configure an Transformation Hub (TH) destination from the SmartConnector with client authentication, but in non-FIPS mode.

Step 1: On the Connector Server

1. Prepare the SmartConnector:
 - **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
 - **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.
2. Navigate to the connector's `current` directory, for example:

```
cd <install dir>/current
```

On Windows platforms:

```
cd <install dir>\current
```

3. Set the environment variables for the static values used by keytool, for example:

```
export CURRENT=<full path to this "current" folder>
export EB=<eb hostname>_<eb port>
export STORES=${CURRENT}/user/agent/stores
export STORE_PASSWD=changeit
export EB_HOST={EB master host IP}
export CA_CERT=/tmp/ca.cert.pem
export INTERMEDIATE_CA_CERT=/tmp/intermediate.cert.pem
export CERT_CA_TMP=/opt/cert_ca_tmp
```

On Windows platforms:

```
set CURRENT=<full path to this "current" folder>
set EB=<eb hostname>_<eb port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=changeit
set EB_HOST={EB master host IP}
set CA_CERT=C:\Temp\ca.cert.pem
set INTERMEDIATE_CA_CERT=C:\Temp\intermediate.cert.pem
set CERT_CA_TMP=/opt/cert_ca_tmp
```

4. Create the **user/agent/stores** directory if it does not already exist, for example:

```
mkdir ${STORES}
```

On Windows platforms:

```
mkdir %STORES%
```

5. Create the connector key pair, for example:

```
jre/bin/keytool -genkeypair -alias ${EB} -keystore
${STORES}/${EB}.keystore.jks -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
```

On Windows platforms:

```
jre\bin\keytool -genkeypair -alias %EB% -keystore
%STORES%\%EB%.keystore.jks -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 365
```

When prompted, enter the password. Note the password; you will need it again in a later step. Press Enter to use the same password for the key.

6. List the key store entries. There should be one private key.

```
jre/bin/keytool -list -keystore ${STORES}/${EB}.keystore.jks -storepass
${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -list -keystore %STORES%\%EB%.keystore.jks -storepass  
%STORE_PASSWD%
```

7. Create a Certificate Signing Request (CSR), for example:

```
jre/bin/keytool -certreq -alias ${EB} -keystore  
${STORES}/${EB}.keystore.jks -file ${STORES}/${EB}-cert-req -storepass  
${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -certreq -alias %EB% -keystore  
%STORES%\%EB%.keystore.jks -file %STORES%\%EB%-cert-req -storepass  
%STORE_PASSWD%
```

Step 2: On the Transformation Hub Server

1. When EB is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in /tmp with the following names:

```
/tmp/intermediate.cert.pem  
/tmp/intermediate.key.pem  
/tmp/ca.cert.pem
```

Use the following command to add them to Transformation Hub:

```
/opt/arcsight/kubernetes/scripts/arcsight-cert-util.sh write --re  
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-  
ca=/tmp/ca.cert.pem
```

Note: If TH was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy TH, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

2.

```
export CA_CERT=/tmp/ca.cert.pem  
export INTERMEDIATE_CA_CERT=/tmp/intermediate.cert.pem  
export INTERMEDIATE_CA_KEY=/tmp/intermediate.key.pem  
export CERT_CA_TMP=/opt/fips_ca_tmp  
export EB=<event broker hostname>_<event broker port>
```
3. Create a temp location on the eb master server:

```
mkdir $CERT_CA_TMP
```

Step 3: On the Connector Server

Copy the `${STORES}/${EB}-cert-req` file (`%STORES%\%EB%-cert-req` on Windows platforms) from the connector to the Transformation Hub directory created above.

Step 4: On the Server

Create the signed certificate, for example:

```
/bin/openssl x509 -req -CA ${INTERMEDIATE_CA_CERT} -CAkey ${INTERMEDIATE_CA_  
KEY} -in ${EB} -cert-req -out ${EB} -cert-signed -days 365 -CAcreateserial -  
sha256
```

Step 5: On the Connector Server

1. Copy the **\${EB}-cert-signed** certificate from the Transformation Hub to the connector's **\${STORES}** directory. (On the Windows platform, copy the **%EB%-cert-signed** certificate to the connector's **%STORES%** directory.)
2. Copy the **ca.cert.pem** certificate from the Transformation Hub to the connector's **\${STORES}** directory. (On the Windows platform, copy the certificate to the **%STORES%** directory.)
3. Copy the **intermediate.cert.pem** certificate from the Transformation Hub to the connector's **\${STORES}** directory. (On the Windows platform, copy the certificate to the **%STORES%** directory.)
4. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot  
-keystore ${STORES}/${EB}.truststore.jks -storepass ${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot  
-keystore %STORES%\%EB%.truststore.jks -storepass %STORE_PASSWD%
```

5. Import the intermediate certificate to the trust store, for example:
- ```
jre/bin/keytool -importcert -file ${STORES}/intermediate.cert.pem -alias
INTCARoot -keystore ${STORES}/${EB}.truststore.jks -storepass ${STORE_
PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\intermediate.cert.pem -
aliasINTCARoot -keystore %STORES%\%EB%.truststore.jks -storepass
%STORE_PASSWD%
```

6. When prompted, enter **yes** to trust the certificate.
7. Import the CA certificate to the key store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -
keystore ${STORES}/${EB}.keystore.jks -storepass ${STORE_PASSWD}
```

**On Windows platforms:**

```
jre\bin\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -
keystore %STORES%\%EB%.keystore.jks -storepass %STORE_PASSWD%
```

8. Import the intermediate certificate to the key store, for example:
- ```
jre/bin/keytool -importcert -file ${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.keystore.jks -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -importcert -file %STORES%\intermediate.cert.pem -alias  
INTCARoot -keystore %STORES%\%EB%.keystore.jks -storepass %STORE_  
PASSWD%
```

If successful, this command will return the message, *Certificate reply was installed in keystore*.

9. When prompted, enter **yes** to trust the certificate.
10. Import the signed certificate to the key store, for example:

```
jre/bin/keytool -importcert -file ${STORES}/${EB}-cert-signed -alias ${EB}  
-keystore ${STORES}/${EB}.keystore.jks -storepass ${STORE_PASSWD}
```

On Windows platforms:

```
jre\bin\keytool -importcert -file %STORES%\%EB%-cert-signed -alias %EB%  
-keystore %STORES%\%EB%.keystore.jks -storepass %STORE_PASSWD%
```

If successful, this command will return the message, *Certificate reply was installed in keystore*.

11. Note the key store and trust store paths:

```
echo ${STORES}/${EB}.truststore.jks  
echo ${STORES}/${EB}.keystore.jks
```

On Windows platforms:

```
echo %STORES%\%EB%.truststore.jks  
echo %STORES%\%EB%.keystore.jks
```

12. Navigate to the **bin** directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

On Windows platforms:

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- a. When completing the Transformation Hub destination fields, use the same values as in Step 8 for the path and password.
 - b. Set **Use SSL/TLS** to **true**.
 - c. Set **Use SSL/TLS Authentication** to **true**.
13. Cleanup. Delete the following files:

Caution: The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive

and should not be distributed to other machines.

```
rm ${STORES}/intermediate.cert.pem  
rm ${STORES}/intermediate.key.pem  
rm ${STORES}/${EB}-cert-signed  
rm ${STORES}/${EB}-cert-req
```

On Windows platforms:

```
del %STORES%\intermediate.cert.pem  
del %STORES%\intermediate.key.pem  
del %STORES%\%EB%-cert-signed  
del %STORES%\%EB%-cert-req
```

Step 6: On the Transformation Hub Server

To cleanup the Transformation Hub server, delete the temporary folder where the certificate was signed and the certificate and key files in `/tmp`.

Caution: The temporary folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

Configure an Transformation Hub Destination without Client Authentication in FIPS Mode

Follow these steps to configure an Transformation Hub destination from the SmartConnector without client authentication in FIPS mode.

On the Transformation Hub Server

When TH is first installed, it's setup to use self-signed certificates. To replace the self-signed certificates, obtain your company's root CA certificate, and an intermediate certificate and key pair. Place them in `/tmp` with the following names:

```
/tmp/intermediate.cert.pem  
/tmp/intermediate.key.pem  
/tmp/ca.cert.pem
```

Use the following command to add them to Transformation Hub:


```
/opt/arcsight/kubernetes/scripts/arcsight-cert-util.sh write --re-  
key=/tmp/intermediate.key.pem --re-crt=/tmp/intermediate.cert.pem --re-  
ca=/tmp/ca.cert.pem
```

Note: If TH was already deployed when running the above command, all Kafka broker nodes must be restarted. Either redeploy TH, or delete each Kafka pod. For example:

To list the pods:

```
kubectl get pods --all-namespaces
```

To delete the first Kafka pod:

```
kubectl delete pod eb-kafka-0 -n eventbroker1
```

Repeat for all the kafka pods.

On the SmartConnector Server

1. Prepare the SmartConnector:

- **If the connector is not yet installed:** Run the installer. After core software has been installed, you will see a window that lets you select **Add a Connector** or **Select Global Parameters**. Check **Select Global Parameters**, and on the window displayed, select **Set FIPS mode**. Set to **Enabled**.
- **If the connector is already installed:** Run the installer. Select **Set Global Parameters** and set **Set FIPS Mode** to **Enabled**.

2. Navigate to the connector's **current** directory, for example:

```
cd <install dir>/current
```

3. Set the environment variables for the static values used by keytool, for example:

```
export CURRENT=<full path to this "current" folder>  
export BC_OPTS="-storetype BCFKS -providername BCFIPS -providerclass  
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath  
{CURRENT}/lib/agent/fips/bc-fips-1.0.0.jar  
-J-Djava.security.egd=file:/dev/urandom"  
export EB=<event broker hostname>_<event broker port>  
export STORES={CURRENT}/user/agent/stores  
export STORE_PASSWD=changeit
```

On Windows platforms:

```
set CURRENT=<full path to this "current" folder>  
set BC_OPTS="-storetype BCFKS -providername BCFIPS  
-J-Djava.ext.dirs=%CURRENT%\jre\lib\ext;%CURRENT%\lib\agent\fips  
-J-Djava.security.properties=%CURRENT%\user\agent\agent.security"
```

```
set EB=<event broker hostname>_<event broker port>  
set STORES=%CURRENT%\user\agent\stores  
set STORE_PASSWD=changeit
```

4. Create the **user/agent/stores** directory if it does not already exist, for example:

```
mkdir ${STORES}
```

On Windows platforms:

```
mkdir %STORES%
```

5. Copy the **ca.cert.pem** and the **intermediate.cert.pem** certificates from the Transformation Hub to the connector's **\${STORES}** directory. (On the Windows platform, copy the certificate to the **%STORES%** directory.)

6. Import the CA certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file ${STORES}/ca.cert.pem -alias  
CARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file %STORES%\ca.cert.pem -alias  
CARoot -keystore %STORES%\%EB%.truststore.bcfips -storepass %STORE_  
PASSWD%
```

7. Import the intermediate certificate to the trust store, for example:

```
jre/bin/keytool ${BC_OPTS} -importcert -file  
${STORES}/intermediate.cert.pem -alias  
INTCARoot -keystore ${STORES}/${EB}.truststore.bcfips -storepass ${STORE_  
PASSWD}
```

On Windows platforms:

```
jre\bin\keytool %BC_OPTS% -importcert -file  
%STORES%\intermediate.cert.pem -alias  
INTCARoot -keystore %STORES%\%EB%.truststore.bcfips -storepass %STORE_  
PASSWD%
```

8. When prompted, enter **yes** to trust the certificate.

9. Note the trust store path:

```
echo ${STORES}/${EB}.truststore.bcfips
```

On Windows platforms:

```
echo %STORES%\%EB%.truststore.bcfips
```

10. Navigate to the **bin** directory and run agent setup. Install a connector with Transformation Hub as the destination, for example:

```
cd <installation dir>/current/bin  
./runagentsetup.sh
```

On Windows platforms:

```
cd <installation dir>\current\bin  
runagentsetup.bat
```

- When completing the Transformation Hub destination fields, use the value from Step 7 for the trust store path and the password used in Step 6 for the trust store password.
 - Set **Use SSL/TLS** to **true**.
 - Set **Use SSL/TLS Authentication** to **false**.
11. Cleanup. Delete the certificate file, for example:

Caution: The following file should be deleted to prevent the distribution of security certificates

that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
rm ${STORES}/ca.cert.pem  
rm ${STORES}/intermediate.cert.pem
```

On Windows platforms:

```
del %\STORES%\ca.cert.pem  
del %\STORES%\intermediate.cert.pem
```

Diagnostic Data and Tools

Transformation Hub includes a diagnostic script (**eb-diag.sh**) for the collection of diagnostic data. Diag.sh is found in the web service container.

To run Transformation Hub diagnostic tools:

1. On the master server, find the Docker container ID of the web service container. (In this example 278e86760803)

```
$ docker ps | grep atlas_web-service
```

```
278e86760803      localhost:5000/arcsightsecurity/atlas_web-  
service@sha256:c25b023afa7b7054de6aa188ed2802d24312f3c5de87b6537aa3e937476376  
d8   "/bin/bash -c 'source'
```

```
$
```

2. Copy the script archive from web service container (In this example 278e86760803)

```
$ docker cp 278e86760803:/eb/ws/eb_diag/eb_diag.tgz .
```

```
$ tar -tzf eb_diag.tgz
```

```
vertica-diag.sh
```

```
eb-diag.sh
```

```
eb-sys-diag.sh
```

```
$
```

3. Extract the diagnostic script and run it

```
$ tar -xvf eb_diag.tgz eb-diag.sh
```

```
$ sh eb-diag.sh
```

Installation Instructions for an Transformation Hub Destination

Master or Worker Nodes Down

This section describes what behavior can be expected if a master node or one or more worker nodes goes down.

- Kubernetes worker nodes will continue running even when the master is down, but if they reboot then they will not be able to find the master and will fail.
- All services running on the master node will become unavailable.
- Transformation Hub Web Service running on the master node becomes unavailable.
 - The services (Routing Stream Process) and integration (ArcMC management) that depend on the Web Service will fail.
 - Any other Transformation Hub (Transform Stream Process, Schema Registry, Kafka Manager) that was running on the master will get scheduled by Kubernetes on other worker nodes depending on system resources available.
 - If the master node was labeled for Kafka and/or ZooKeeper deployment, then those instances will fail but the cluster will still work with rest of the instances on worker nodes
- The NFS server, which runs on the master node, will become unavailable.
 - Kafka and ZooKeeper do not depend on NFS storage and use local Kubernetes worker node storage. They would still be available for event processing with some limitation.
 - The beta feature Connector in TB (CTB) will be affected, since it depends on NFS storage, which is configured on master server.
- DNS service (**kube-dns**) runs on the master server will become unavailable
 - Worker nodes would lose ability to resolve hostnames, except for those that had already been resolved, and which may be cached for some period.
- Any of the Transformation Hub service instances running on the downed worker node that are not tied to a worker node (such as Transform Stream Process, Routing Stream Process, Schema Registry, or Kafka Manager) will be scheduled by Kubernetes on other worker nodes, depending on system resources available on other worker nodes.
- Depending on system resources on other worker nodes, Transformation Hub service instances that are labeled for Kafka and ZooKeeper will be automatically scheduled by Kubernetes on other worker nodes

(as long as there are additional worker nodes already labeled for Zookeeper and kafka).

- Likewise, the c2av-processor may cease if the worker node containing the **eb-c2av-processor** goes down and system resources prevent Kubernetes from automatically rescheduling processing on another worker node.
- If automatic re-scheduling of service instances does not occur for the Zookeeper, Kafka, or **eb-c2av-processor** (that is, the node is not recoverable), use the following manual command, run from the master node, to delete all service instances from the failed node and force Kubernetes to move the services to other nodes:

```
# kubectl delete node <Failed_Node_IP>
```

Note: there must be another node available in the cluster, with the **zookeeper** and **kafka** labels, for the service instances to be migrated from the failed node.

Appendix C: Troubleshooting

These troubleshooting tips may prove helpful in resolving issues with Transformation Hub.

This chapter includes the following topics:

- [Self-Healing for Unparsed Events](#)151
- [SmartConnector Commands Queue](#)152
- [Tuning Transformation Hub Performance](#)152
- [Verifying the Health of the Transformation Hub Cluster](#)153
- [What is FIPS?](#) 154

Self-Healing for Unparsed Events

If the **Generate Unparsed Events** feature is on, and there is high traffic of events, the destinations might receive less valuable information (unparsed events).

Users can now set a limit for unparsed events. When the number of unparsed events reaches the limit, the **Self-Healing** feature disables the unparsed event generation and as soon as the event queue normalizes, it is re-enabled and the destinations start receiving unparsed events again.

If the self-healing feature is active, an internal event is sent with the status of the **Generate Unparsed Events** feature.

New Properties added

- **unparsed.events.self.healing.enabled** —the default value is False. Set to **True** to turn on the functionality.
- **unparsed.events.self.healing.threshold.limit**—the default value is 60%. The functionality is enabled when the limit is reached.

Note: The limit refers the percentage of unparsed events in a given time

To enable Self-Healing

From the **agent.properties** file, ensure the **unparsed.events.self.healing.enabled** flag in is set to True.

SmartConnector Commands Queue

Commands Queue prevents from executing commands that might conflict if executed at the same time. Conflicting commands are sent to a queue and executed later.

The ESM Console can track commands sent to an agent. When the console finds a conflict, it does not send the command. However, if another console sends the command to the agent at the same time, there might be a conflict among each other.

Some command groups might present an error when executed from different ESM Consoles in parallel processes, or, if the same command is executed more than once, at the same time.

Commands received by a connector are now placed in a queue and categorized, according to their priority.

Command Type	Description
"Get" Commands	Always safe to execute Skip the queue Executed at the moment they arrive.
Any other commands	Low conflict probability, unless another process is running the same command.
Event Flow Commands	Change the connector status (start, stop, pause).
Upgrade Commands	Change the connector version.
Connector Process Commands	Turn off the connector (terminate, restart). Its priority can be changed to always choose these commands above the others To change the priority: <ol style="list-style-type: none">1. Go to the agent.properties file.2. Add the property commands.queue.connector.process.high.priority=true.3. Ensure the property status is set to True.

Tuning Transformation Hub Performance

The following can help improve the performance of Transformation Hub.

Increasing Stream Processor EPS

You can increase Stream Processor EPS by adding more stream processor instances using the ArcSight installer configuration UI.

When you change this value, you do not need to redeploy Transformation Hub. Please note that this change will increase the number of pods. You will see this difference when you run the `kubectl get pods --all-namespaces` command.

Increasing Kafka Retention Size/Time

You can change the value of retention size or time in any topic using Transformation Hub Manager after deploying Transformation Hub containers, and it will be applied immediately. You can change this while events are flowing through the topic.

To change the default values before you deploy, change the values in the `arcsight-installer.properties` file.

Adding a New Worker Node

To add a new worker node, label the new node (delete or overwrite existing label with a different label). Remove the label from the old node. Kubernetes should start Kafka on the new node. Then, reassign partitions on the new node. Data copying will take some time to complete.

Verifying the Health of the Transformation Hub Cluster

Verify the health of each container: run `#kubectl get pods --all-namespaces -o wide` to list pods and their status.

View Kubernetes logs for each container: run `kubectl logs`

`# kubectl logs -n eventbroker1 [WEB SERVICE POD ID/NAME] -c atlas-web-service`

Verify data flows through the system: check any of the following.

- In ArcMC, review the EPS graph. This indicates whether events are flowing through the stream processor (routing and transforming).
- All topics: check the offset for each topic in Transformation Hub Manager. you should see the value increasing.

Verify that Web Service APIs are healthy:

- Check logs of the web service container (see command above)
- Make sure the port is bound:

```
# netstat -lntp | grep 38080
```

- Check the Kafka Scheduler status.

```
# watch ./root/install-vertica/kafka_scheduler status
```

- Check whether the offset is increasing in the status output. If not, then there may no data in the Avro topic, or if Avro contains data there may be a problem.
- Verify the topic partition count and distribution.
- Check that the configured partition count matches its expected value
- Check the partition count or replication factor for the topic using Transformation Hub Manager.

What is FIPS?

Under the Information Technology Management Reform Act (Public Law 104-106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for Federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use government-wide. NIST develops FIPS when there are compelling Federal government requirements such as for security and interoperability and there are no acceptable industry standards or solutions.

FIPS Suite B includes cryptographic algorithms for hashing, digital signatures, and key exchange. The entire suite of cryptographic algorithms is intended to protect both classified and unclassified national security systems and information.

Note: When FIPS-compliant connectors connect to a non-FIPS-compliant destination, the solution is not considered FIPS compliant. When the destination is installed in FIPS Suite B compliant mode, the SmartConnectors must also be installed in FIPS Suite B compliant mode.

Appendix D: Connector Frequently Asked Questions

The following are a list of frequently asked questions. This section is periodically updated.

- [My machine is in a different location than 'en_US' and my connectors are running into parser errors when parsing timestamp fields.](#)
- [What if my device is not one of the listed connectors?](#)
- [My device is on the list of supported products; why doesn't it appear in the connector Configuration Wizard?](#)
- [Why isn't the SmartConnector reporting all events?](#)
- [Why are some event fields not showing up in the Console?](#)
- [Why isn't the SmartConnector reporting events?](#)
- [How can I get my database SmartConnector to start reading events from the beginning?](#)
- [When events are cached and the connection to the Manager is re-established, which events are sent?](#)
- [Why does the status report the size of the cache as smaller than it should be? For example, I know that a few events have been received by the SmartConnector since the Manager went down, yet the report marks events as zero.](#)
- [Why does the estimated cache size never change in some connectors? Why is the estimated cache size negative in others?](#)
- [Can the SmartConnector cache reside somewhere other than user/agent/agentdata?](#)
- [Why is my end time always set to a later date and time?](#)
- [Do our Syslog connectors support forwarded messages from KIWI or AIX?](#)
- [What does the T mean in the periodic SmartConnector status lines?](#)
- [What do Evts and Eps refer to?](#)
- [Does a file reader SmartConnector reading files over a network share display errors when the network share is disconnected? How can I recognize which error message refers to which file in agent.log and agent.out.wrapper.log?](#)
- [Are log files accessed sequentially or in parallel?](#)
- [After reading a log file, can a SmartConnector move them using NFS?](#)
- [My SmartConnector must read log files from a remote machine through a network share. How can I do this?](#)
- [Is there any limitation on performance relating to EPS?](#)
- [How many log files can a SmartConnector access at one time?](#)

- Which is the default cache size limit?
- How the cached DNS entries are managed? If there is a size limit, how would it be handled? Will it rotate when the limit exceeds?
- Under what circumstances can an entry be removed of the cache?
- How should an entry be manually removed from the cache?
- What is the recommended maximum number of connectors per Manager?
- When configuring the connector to run as a service (for Windows) or daemon (for Unix), you may encounter the following error message: An issue has been encountered configuring the connector to run as a service. Check agent.log (Service Installation) for details.

My machine is in a different location than 'en_US' and my connectors are running into parser errors when parsing timestamp fields.

The connector assumes a default locale of 'en_US'. If your machine is running in a different locale, your connector may run into parsing errors when parsing timestamps. Try changing the parser locale by adding a property '`agent.parser.locale.name=<locale of your machine>`' into `user/agent/agent.properties`, then restart your connector.

For example, China and France would have the following locales:

```
agent.parser.locale.name=zh_CN
agent.parser.locale.name=fr_FR
```

To use the default locale for the connector machine, you can leave the locale blank. For example:

```
agent.parser.locale.name=
```

What if my device is not one of the listed connectors?

- ArcSight offers an optional feature called the FlexConnector Development Kit (SDK), which can assist you in creating a custom connector for your device.
- ArcSight can create a custom connector; contact customer support for more information.

My device is on the list of supported products; why doesn't it appear in the connector Configuration Wizard?

connectors are installable based upon the operating system you are using. If your device is not listed, either it is not supported by the operating system on which you are attempting to install, or your device is served by a Syslog server and is, therefore, a syslog sub-connector. To install a Syslog connector, select **Syslog Daemon**, **Syslog Pipe**, or **Syslog File** during the installation process.

Why isn't the SmartConnector reporting all events?

Check that event filtering and aggregation setup is appropriate for your needs.

Why are some event fields not showing up in the Console?

Check that the two separate turbo modes for the connector and the Manager are compatible for the specific connector resource. If the Manager is set for a faster turbo mode than the connector, some event details will be lost. See ["Understanding ArcSight Turbo Modes" on page 29](#) for detailed information.

Why isn't the SmartConnector reporting events?

Check the connector log for errors. Also, if the connector cannot communicate with the Manager, it caches events until its cache is full. A full cache can result in the permanent loss of events.

How can I get my database SmartConnector to start reading events from the beginning?

If it is a FlexConnector for Time-Based DB, set the following parameter in agent.properties:

```
agents[0].startatdate=01/01/1970 00:00:00
```

If it is an FlexConnector for ID-Based DB, set the following parameter in agent.properties:

```
agents[0].startatid=0
```

When events are cached and the connection to the Manager is re-established, which events are sent?

Events are sent with a 70% live and 30% cached events ratio. If live events are not arriving quickly, the percentage of cached events can be higher. This can reach 100% if there are no live events.

Also, if the settings dictate that certain event severities are not sent at the time connection is restored, those events are never sent. This is true even if they were originally generated (and cached) at a time when they would ordinarily go out.

Why does the status report the size of the cache as smaller than it should be? For example, I know that a few events have been received by the SmartConnector since the Manager went down, yet the report marks events as zero.

Some of the events are in other places in the system, such as the HTTP transport queue. Shut down the connector and look at the cache size in the .size.dflt file to confirm that the events are really still there.

Why does the estimated cache size never change in some connectors? Why is the estimated cache size negative in others?

The estimated cache size is derived from a size file that gets read at startup and written at shutdown. If the connector could not write the size at shutdown (for example, due to an ungraceful shutdown, disk problem, or similar problem) the number could be incorrect. Newer versions will attempt to rebuild this cache size if they find it to be incorrect, but older builds do not.

One solution is to:

1. Stop the connector.
2. Delete the size file (a file with extension `.size.dflt`) under `current\user\agent\agentdata`.
3. Re-start the connector.

The connector detects that there is no size file and re-builds the cache size by reading all the cache files.

Can the SmartConnector cache reside somewhere other than user/agent/agentdata?

You can change the folder to contain the connector cache by adding the following property in `agent.properties`:

`agentcache.base.folder=<relative-folder-path>`

where `<relative-folder-path>` is the path of the folder relative to `$ARCSIGHT_HOME`.

Why is my end time always set to a later date and time?

The Manager performs auto time correction for older events. If the end time is older than your retention period, it is set automatically to that lower bound. A warning is displayed and an internal event with the same message is sent to you.

Do our Syslog connectors support forwarded messages from KIWI or AIX?

Yes.

The property related to KIWI is

`syslog.kiwi.forwarded.prefix=KiwiSyslog Original Address`

Kiwi adds a prefix with the original address. For example, the message:

`Jan 01 10:00:00 myhostname SSH connection open to 1.1.1.1`

is converted to

`Jan 01 10:00:00 myhostname KiwiSyslog Original Address myoriginalhost: SSH connection open to 1.1.1.1`

The connector strips out the prefix and uses `myoriginalhost` as the Device Host Name.

The property related to AIX is

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

Similar actions are performed for messages forwarded using AIX.

What does the T mean in the periodic SmartConnector status lines?

"T" is shorthand for "throughout (SLC)." The following lines are in `agent.defaults.properties`:

```
status.watermark.stdoutkeys=AgentName,Events  
Processed,Events/Sec(SLC),Estimated Cache  
Size,status,throughout(SLC),hbstatus,sent  
status.watermark.stdoutkeys.alias=N,Evts,Eps,C,ET,T,HT,S
```

The SLC stands for Since Last Check, which means "in the last minute," assuming `status.watermark.sleepTime=60` has not been overridden.

What do Evts and Eps refer to?

Evts is an acronym for Events Processed and **Eps** is an acronym for Events/Sec(SLC).

Does a file reader SmartConnector reading files over a network share display errors when the network share is disconnected? How can I recognize which error message refers to which file in agent.log and agent.out.wrapper.log?

If the network share is a Linux/UNIX NFS mount or a Windows network mapped drive, the file reader connector displays errors in the agent log.

If files are being read using a Windows UNC path that does not require network mapping, the file reader connector cannot detect a network connection loss.

Error messages related to file access contain the file name, but error messages related to log line parsing do not.

Are log files accessed sequentially or in parallel?

This depends upon the connector you are using. Some log file connectors process files sequentially and others process log files in parallel.

After reading a log file, can a SmartConnector move them using NFS?

Yes. Folder Follower connectors can rename or move the files using NFS, as long as the folders containing the log files give the correct permissions for the connector.

My SmartConnector must read log files from a remote machine through a network share. How can I do this?

To establish a network share to a remote machine, you can use network mapping on Windows platforms, and NFS or Samba mounting on Linux/UNIX platforms.

If you are running the connector as a Windows service, access privileges to the network share are required. To access the user name and password panel:

1. From the **Start** menu, select **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Services**.
4. Right-click the name of the appropriate connector and select **Properties**.
5. Click the **Log on** tab, and enter the user name and password for the user with access permissions to the file share. Specify the file path using UNC notation, not as a network mapped drive.

Is there any limitation on performance relating to EPS?

These limitations are subjective and depend upon system resources, number of devices, number of events, and so on.

How many log files can a SmartConnector access at one time?

The connector can access as many log files as it is configured with. The folders are processed in parallel.

What is the recommended maximum number of connectors per Manager?

There is no hard and fast maximum. The Manager has a restriction of 64 concurrent connector threads by default. The more threads you add, the more it affects performance, because there is more thread context-switching overhead. The general recommendation is to definitely stay lower than the triple-digit range.

When configuring the connector to run as a service (for Windows) or daemon (for Unix), you may encounter the following error message: An issue has been encountered configuring the connector to run as a service. Check agent.log (Service Installation) for details.

There may be different reasons for you to get this message when you cannot configure the connector to run as a service or daemon. It may be that you installed a second connector on Windows or Unix with the exact same name and type, such as when using the default options. More information is included in the *agent.log*, including the specifics for <Service Installation>. For example, <Service Installation> - SE:wrapperm | Unable to install the ArcSight Syslog NG Daemon service - The specified service already exists. (0x431).

You can fix this issue by manually deleting the `agent.wrapper.conf` file from the second or additional connectors. The file is located in the `$ARCSIGHT_HOME/current/user/agent` folder.

When configuring multiple connectors, use a different name and type to avoid duplication.

Which is the default cache size limit?

SmartConnectors use a compressed disk cache to hold large volumes of events when the destination (such as ArcSight Manager) is down or when the SmartConnector receives bursts of events. This parameter specifies the disk space to use. The default is 1 GB which, depending on the connector, can hold about 15 million events (though this can vary dramatically depending on the event type), but it also can go down to 200 MB. When this disk space is full, the SmartConnector drops the oldest events to free up disk cache space. Select from the options available in the dropdown list. Default is 1 GB.

How the cached DNS entries are managed? If there is a size limit, how would it be handled? Will it rotate when the limit exceeds?

If the Name Resolution Host Name Only parameter is set to No, then the source, destination, and device host name and DNS domain event fields are looked at. If in any of the three cases the DNS domain event field is empty and the corresponding host name event field contains a host name that is not an IP address (IPv4 or IPv6) and does have a dot in it, then the host name event field is split at the first dot, with the latter part being moved to the corresponding DNS domain event field.

If, on the other hand, the corresponding DNS domain and host name event fields are both set for the source, destination, or device, and the host name event field does not contain any dots, then they are combined for purposes of any name resolution that may be done later.

Next, if the Name Resolution Host Name Only parameter is set to No, and the agent host name event field is set (which is normally done for all Connectors based on a lookup done at startup time), contains a dot, and is not an IP address, then it is split up, with the part after the first dot being moved to the agent DNS domain event field.

Lastly, if the Name Resolution Domain From E-mail parameter is set to Yes, then the source and destination DNS domain event fields are looked at. If either is empty, the corresponding user name event field is not empty and contains an "@" character, and the corresponding host name event field is empty, then the DNS domain event field is set to the part of the user name event field after the "@" character.

On the other hand, the default limit is 50000 entries for each of the two caches (names => IPs, and IPs => names). An expiration daemon runs periodically (normally once per minute) to check for stale cache entries, as defined by the shortest TTL set for all destinations. Older entries are simply queued for re-resolution. If the queues are at all backed up, these refresh requests may take some time, which is why entries that are up to twice the TTL are not considered stale when Wait For Resolution is disabled. Note that previously, arbitrarily old entries would continue to be used, so the behavior in the case of severely backed up queues has changed.

Under what circumstances can an entry be removed of the cache?

Cache entries are removed if 1) the size limit is reached or 2) the name or IP address is explicitly configured to not be looked up but it was previously in the cache. Additionally, an entry can be removed after DNS lookup fails if it has been previously configured by removing Unresolvable Names /Ips from Cache.

How should an entry be manually removed from the cache?

It is not possible to manually remove a single entry from the cache. It is only possible to flush the cache files while the connector is stopped. The action must not be documented.

Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on SmartConnector User Guide (Connectors 7.13.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to arcsight_doc@microfocus.com.

We appreciate your feedback!