Micro Focus Security ArcSight SmartConnectors

SmartConnector for Microsoft Windows Event Log

Configuration Guide

Document Release Date: April 5, 2019
Software Release Date: April 5, 2019

Legal Notices

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

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://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Chapter 1: Product Overview

The infrastructure provided with the SmartConnector for Microsoft Windows Event Log has been improved to deliver critical features such as Operational Windows Event Logs and event collection and event filtering from IPv6 hosts. It leverages the native technology on the Microsoft platform and provides the best support for Windows event features and capabilities (including collection for all log types).

Note: Security events are not audited by default. Be sure to specify the type of security events to be audited (see "'Enable Microsoft Windows Event Log Audit Policies" on page 11" in this document).

This connector consists of three major components:

- SmartConnector framework-based event processor
- The Windows Eventing API, which collects events from Microsoft Windows Event Logs
- A Message Queue that facilitates communication between the previous two components

The Windows API event collection and the Message Queue are started by the connector at the time of connector setup and at the start of the connector process.

Chapter 2: Introduction

This guide provides information for installing the SmartConnector for Microsoft Windows Event Log and configuring the device for event-log collection.

Installation platforms:

- CentOS 6.9 and 7.4
- RHEL 6.9 and 7.4
- ArcMC 2.80
- ArcMC 2.81

ArcSight SmartConnectors provide easy, scalable, audit-quality collection of all logs from all event-generating sources across the enterprise for real-time and forensic analysis.

For SmartConnector security event mappings to ArcSight data fields, see SmartConnector for Microsoft Windows Event Log Windows Security Event Mappings.

Chapter 3: Features

SmartConnector capabilities include real-time event collection and processing, as well as data enrichment (normalization, categorization, Common Event Format (CEF), aggregation, and filtering) and efficiency (caching, batching, compression, and bandwidth management). For more information, see the ArcSight *SmartConnector User Guide*. Specific features of the Windows Event Log connector are described in the following sections.

Custom Log Support

Event collection from non-administrative, operational, or custom logs is provided.

Event Filtering

Filters that apply at the time of event collection from the event source to the connector are supported. With this support, events in which you have no interest can be filtered out, making better use of resources.

Globally Unique Identifier (GUID)

Translation and mapping of the GUID (also known as UUID) within a forest is supported. (A forest is a complete instance of Active Directory.) The connector can perform GUID translation for GUIDs within a forest by querying the Global Catalog Server. The Active Directory parameters are used for Global Catalog Server. The connector is not configured to translate GUIDs by default. See "Advanced Configuration Parameters for SID and GUID Translation" for more information about enabling GUID translation. Global Catalog and Active Directory must be on the same machine.

Host Browsing

Host browsing is used when hosts are added during installation using Active Directory. Notification is sent to a destination when a new host is added to Active Directory.

IPv6

Event collection from IPv6 hosts and parsing of IPv6 events is supported.

Operating Systems Supported for Event Collection

SmartConnectors supports Windows Event Log Security, System, and Application event collection from hosts running the following Microsoft OS versions.

- Microsoft Windows Server 2008 R2 SP1
- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows 7 (Service Pack 1)
- Microsoft Windows 8
- Microsoft Windows 10

It also supports events forwarded from source hosts to a Windows Event Collector (WEC).

Installation Requirements

System Requirements

The connector can be installed on one of the following Linux-based platforms:

- CentOS 6.9 and 7.4
- RHEL 6.9 and 7.4
- ArcMC 2.80
- ArcMC 2.81

Events Supported

Windows Event Log supports parsing for:

Event Type	Event Header	Event Description
Security	yes	yes
Application	yes	no*
System (Service Control Manager and WINS event sources)	yes	yes
Other System events (including Remote Access and NPS)	yes	no*

* Support is provided for a Flex-Connector-like framework that lets you create and deploy your own parsers to parse the event description for all system and application events. See "Create and Deploy Parsers for System and Application Events" for more information. See "Log Parser Support" for application and system events already supported.

Log Parser Support

The SmartConnector supports parsing for the following logs:

- Security
- System
- Application (event header)
- Forwarded Events (for forwarded security, system, and application (event Header) events

Applications Supported

Parser support for the following application events is provided:

- Microsoft Active Directory
- Microsoft Exchange Access Auditing
- Microsoft Forefront Protection 2010
- Microsoft SQL Server Audit
- Oracle Audit
- Symantec Mail Security for Exchange

System Events Supported

Parser support for the following system events is provided:

- Microsoft Network Policy Server
- Microsoft Remote Access
- Microsoft Service Control Manager
- Microsoft WINS Server

Use of Active Directory Query for Hosts

An Active Directory query can be used to populate or update collection end points, or to specify the Windows OS version of source hosts for forwarded events if collected from the Windows Event Collector. The connector discovers and retrieves information about the hosts registered in an Active Directory. The host information includes the DNS name along with its operating system version. When new hosts are registered in an Active Directory while the connector is running, it sends an internal event notifying the user of the newly discovered host.

Chapter 4: Configure Windows

Enable Microsoft Windows Event Log Audit Policies

Because event information generated by Windows servers is based upon which auditing policies are enabled, ensure the appropriate auditing policies are enabled on those Windows servers from which the connector will be collecting information. By default, none of the Windows auditing features are turned on.

When planning which events to audit, keep in mind that auditing events consumes system resources such as memory, processing power, and disk space. The more events you audit, the more of these resources are consumed. Auditing an excessive number of events can dramatically slow down your servers.

Note: You must be logged on as an administrator or a member of the Administrators group to set up audit policies. If your computer is connected to a network, network policy settings might also prevent you from setting up audit policies.

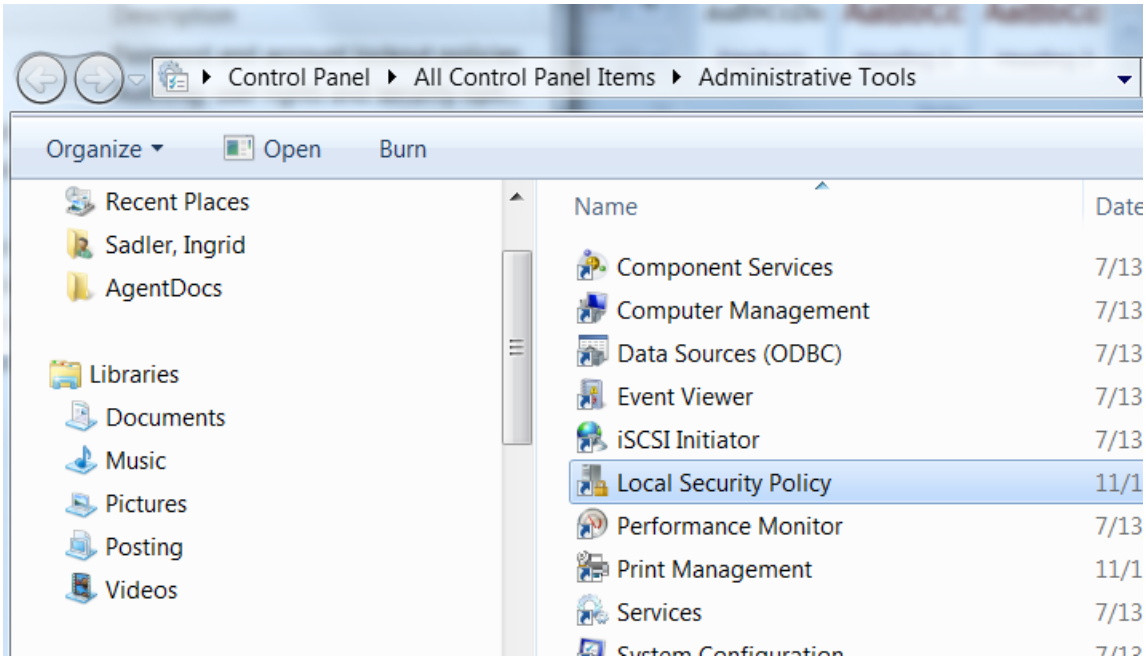
The method used to create an audit policy varies slightly depending upon whether the policy is being created on a member server, a domain controller, or a stand-alone server.

- To configure a domain controller, member server, or workstation, use **Active Directory Users and Computers**.
- To configure a system that does not participate in a domain, use **Local Security Settings**.

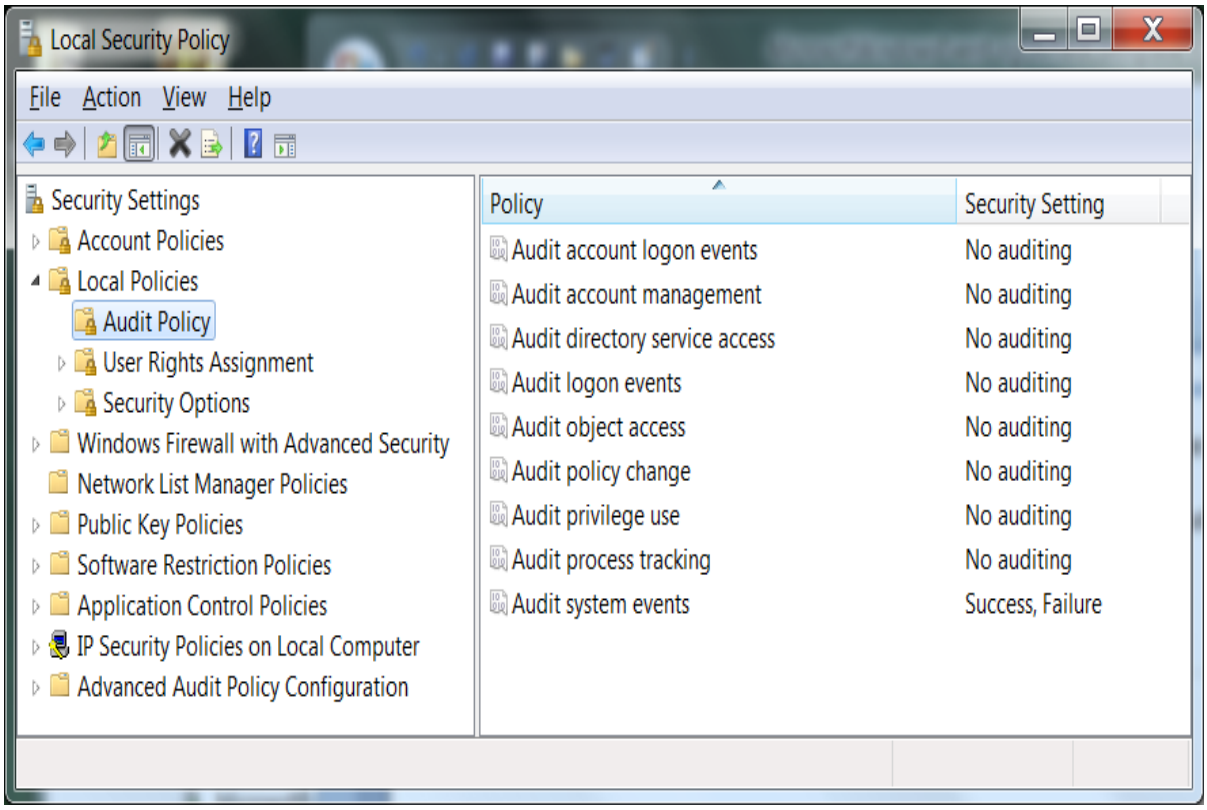
Audit a Local System

To establish an audit policy on a local system:

1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.

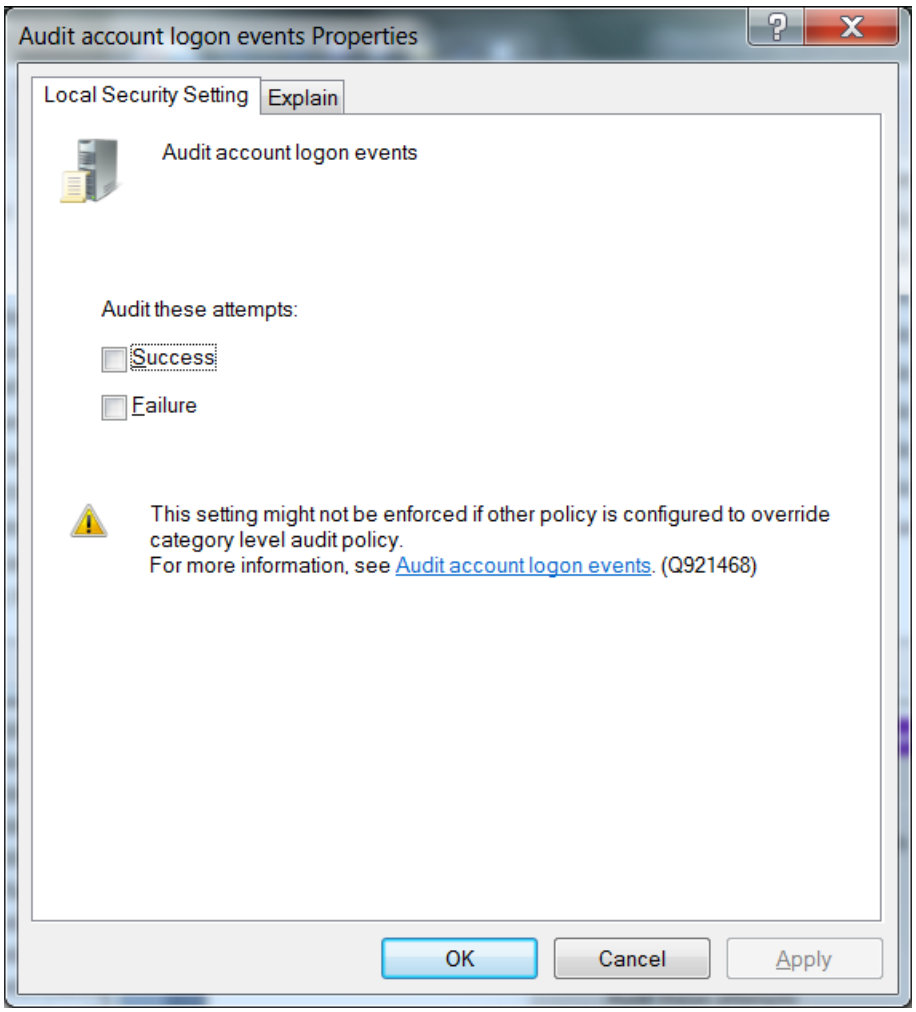


2. Double-click on **Local Policy** in the **Security Settings** tree to expand it.
3. Select **Audit Policy** from the tree. Doing so reveals the auditing information for that system.



4. To enable auditing for any of the areas, double-click on the type of audit; a dialog box such

as the following is displayed, letting you choose to perform a **Success** or a **Failure** audit (or both) on that type of event.



Note: To audit objects such as the Registry, printers, files, or folders, select the Object Access option. Otherwise, when you attempt to enable auditing for these objects, an error is displayed instructing you to make the necessary adjustments to the local audit policy (or, in the case of a domain environment, to the domain audit policy).

Once you have enabled auditing, go through the system and fine-tune the type of events that will be audited in each category.

Set Up an Audit Policy Within a Domain

To set up an audit policy for a domain controller:

1. Choose **Start > Programs > Administrative Tools > Active Directory Users and Computers**.
2. Navigate through the console tree to the domain you want to work with. Expand the domain.

3. Beneath the domain, you will see a **Computers** object and a **Domain Controllers** object. Select the appropriate object for your system and right-click on **Domain Controllers**. The Domain Controller's properties sheet is displayed.
4. Select the **Group Policy** tab. Select the group policy to which you want to apply the audit policy and click **Edit**.
5. Navigate through the tree to **Default Domain Controllers Policy > Computer Configuration > Windows Settings > Security Settings Local Policies > Audit Policy**.
6. When you select **Audit Policy**, a list of audit events is displayed in the right pane. To audit a group of events, double-click on the group; a dialog box is displayed that lets you enable **Success**, **Failure**, or both audits for that group of events.

After enabling auditing for a group of events, fine-tune the exact events you want to audit.

Set Up an Audit Policy for a Domain

To set up auditing for all computers under a domain:

1. Click **Start > Administrative Tools > Domain Security Policy**.
2. Open **Default Domain Security Settings**.
3. Expand **Security Settings** if it is not already open.
4. Expand **Local Policy** and double-click on **Audit Policy**. A list of audit events is displayed in the right pane.
5. To audit a group of events, double-click on the group; a dialog box is displayed that lets you enable **Success**, **Failure**, or both audits for that group of events.

Windows Host Prerequisites

- NET Framework

  The minimum .NET version requirement is 4.5.
- PowerShell

  The minimum PowerShell version requirement is 5, and PowerShell Remoting needs to be enabled. Some Windows Versions such as Windows 10 and Windows Server 2016 have Powershell 5.0 or above already available but the older Windows Versions might not have this PowerShell minimum version installed, in this scenario, some Windows versions are elegible for installing a WMF (Windows Management Framework) package on them, this package contains some important functionalities, including PowerShell 5.0 or above. Please verify if the audited Windows host without the minimum required PowerShell version is elegible for this WMF installation. For further reference please refer to the Microsoft official documentation on it. If it is not eligible, you can leverage windows event forwarding, to forward logs from such hosts to

a central collector host, and configure the connector to collect events from a central collector host.

Configure Event Collection Permissions

Using Administrator User Account

Local Administrator users do not need special configurations in order to use them to audit the Windows host using the connector, the same can be said for Domain Administrator users as long as they have administrative privileges on the audited machine.

When using an Administrator account just follow the steps to check if the WinRM service is running or not and the steps to enable either the HTTP or the HTTPS listener.

Using Domain Standard User Account

Standard Domain Users can be added individually on each audited machine to the correct groups described on the Standard Local Users section, however, in order to save time and effort a Policy can be created on the Domain in order to automatically add the Standard Domain User to the Event Log Readers local group in each machine of the domain, to do this just follow the next steps:

1. Log into the Domain Controller machine and open: **Control Panel > Administrative Tools > Group Policy Management** and expand the desired Domain for the Policy creation.
2. Right-click **Default Domain Policy** and click **Edit**. (A new policy can be created as well).
3. From the emerging window go to: **Default Domain Policy > Computer Configuration > Preferences > Control Panel Settings**, right-click **Local Users and Groups** and choose **New > Local Group**.
4. From the new emerging window on the tab set the next values:
   a. Action: Update
   b. Group name: The group where the user is going to be added on each machine of the domain, **Event Log Readers**.
   c. Click on add button and choose the Standard Domain User that is going to be included on each local group of the machines in the domain.
   d. Click **Apply** and then **OK** to accept the values and close the window.

   This Group Policy can take some time to take effect.
5. To enable the policy immediately, run this command from the Windows Server Domain

Controller and the Windows Members command prompts:

```
GPUpdate /Force
```

Note: The `GPUpdate /Force` command will update any modifications you have made to any group policy, not just this one.

The addition of the Standard Domain User account to the Remote Management Users group is done manually on each machine of the domain, in order to do this just log into the desired machine and go to **Control Panel > Administrative Tools > Computer Management > System Tools > Local Users and Groups > Groups** and add the Standard Domain User to the Remote Management Users group (or WinRMRemoteWMIUsers_ for older operating systems).

Note: If the audited machine is a domain controller then the user has to be added to: **Control Panel > Administrative Tools > Active Directory Users and Computers >(Domain Name) > Builtin**

Note: If the Remote Management Users group is not available on older Windows versions the group WinRMRemoteWMIUsers_ can be used instead, if neither of both groups are present then a regular group can be created manually and add the user to it.

Note: To implement a standard local user account from Windows workgroup hosts:

1. Go to **Settings > Control Panel > Administrative Tools > Local Security Policy > Security Settings > Local Policies > Security Options**.
2. Open the **Network access: Sharing and security model** for local accounts policy.
3. Set this policy to the option: **Classic – local users authenticate as themselves**.

Using Local Standard User Account

Take the desired Local user and add it to the correct groups, to do this, go to Control Panel > Administrative Tools > Computer Management > System Tools > Local Users and Groups > Groups and add the user to the next two groups: Event Log Readers and Remote Management Users.

Note: If the Remote Management Users group is not available on older Windows versions the group WinRMRemoteWMIUsers_ can be used instead, if neither of both groups are present then a regular group can be created manually and add the user to it.

Set Up Standard User Accounts

The connector does not require domain administrator privileges to collect Security events from Windows hosts. Event Log Reader privilege is required for system and custom application event collection (including Forwarded Events Collection).

To configure the SmartConnector for Microsoft Windows Event Log to use a Standard User account to collect Security events only from the target hosts, follow the steps provided in the following sections.

These steps describe how to configure and assign the privileges by creating a single user account such as **arcsight**. You can also create a group of users instead and follow the same steps provided for the configuration, assigning all the minimum privileges to the user group instead of the single user.

Note: Sometimes, although we have assigned appropriate privileges to the standard user, there could be other policies in your environment preventing the user account from accessing the security event logs. You can start identifying this problem by checking **Settings > Control Panel > Administrative Tools > Local Security Policy > Security Settings > Local Policies > Security** options. There are many security policies defined that would require investigation; however, one policy to check right away is the **Network Access: Sharing and security model for local accounts**. Make sure this is set to **Classic – local users authenticate as themselves**.

Standard Domain User Account from Domain Members

On the Windows Server Domain Controller:

1. Go to **Settings > Control Panel > Administrative Tools > Active Directory Users and Computers > <Domain of interest> > Users**.
2. Create a new Domain User, such as `arcsight`.
3. Go to **Settings > Control Panel > Administrative Tools > Group Policy Management > Default Domain Policy > Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
4. Open the **Manage auditing and security log** policy.
5. Enable **Define these Policy Settings** and add this new Domain User `arcsight` to this policy.
6. This Group Policy can take some time to take effect. To enable the policy immediately, run this command from the Windows Server Domain Controller and the Windows Domain Member command prompts:

   ```
   GPUpdate /Force
   ```

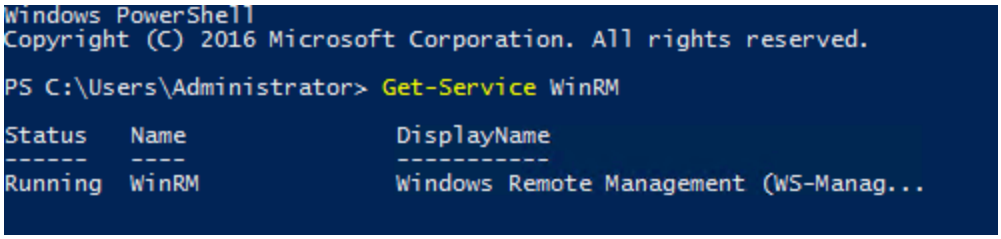
Note: This command will update modifications to any group policy you have made, not just this one

WinRM Configuration

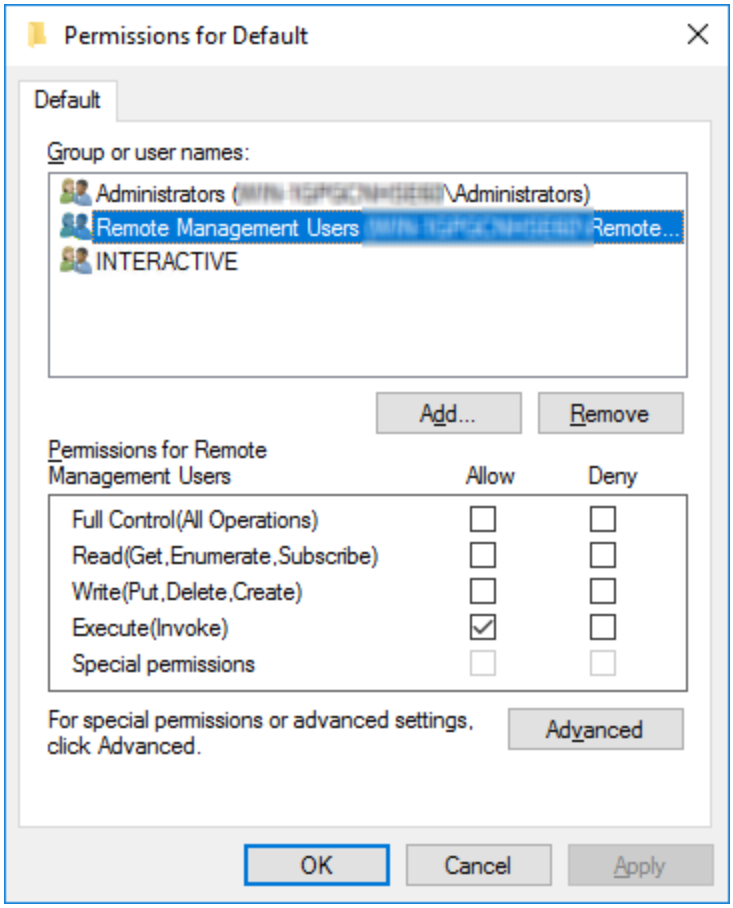
The SmartConnector supports connections to the configured Windows hosts using NTLM authentication over either an HTTP or HTTPS listener.

1. In order to make these scenarios work, run the `Get-Service WinRM` command to ensure that the WinRM service is running on the server machine.

   If the command returns the following result, then the WinRM service is running.

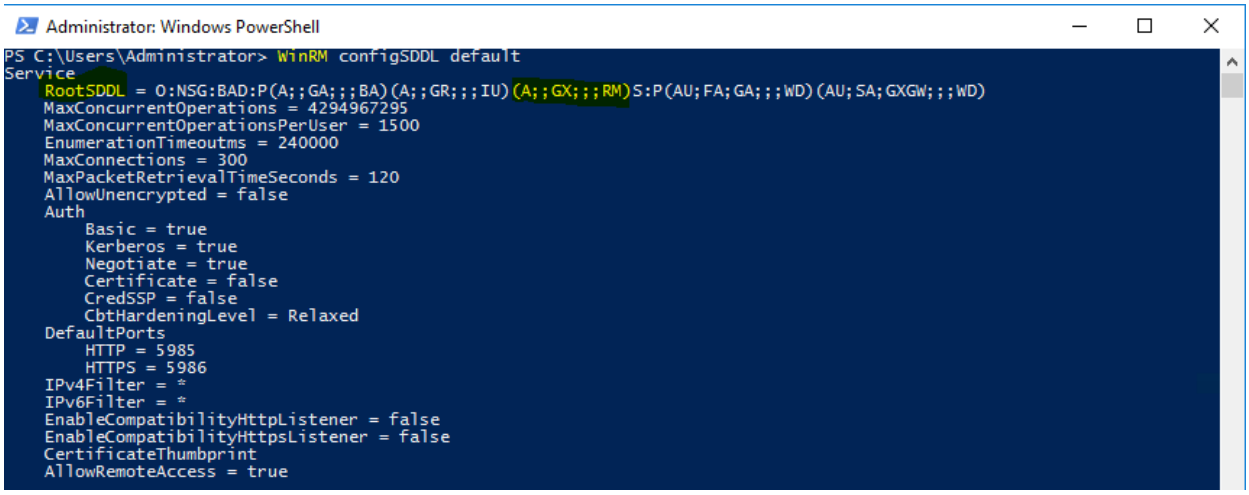
   

   If the WinRM service is not running, then run the `WinRM QuickConfig` command to enable the WinRM service with its default parameters.
2. After checking again that the WinRM service is running correctly, configure the Standard User Account for remote use in this server.
3. Add the **Remote Management Group** (or the chosen group) to the rootSDDL of the machine.
4. To open the permissions panel, run the `WinRM configSDDL default` command from a PowerShell as administrator.
5. Add the **Remote Management Users** (or the chosen group) from that machine and assign it only the execute permissions, shown below.



6. Once the addition is complete, click **OK** to close the dialog.

   A message with the SDDL appears on the console showing the newly configured parameters.



7. Check the RootSDDL line.

It should contain: `(A;;GX;;RM)`. This indicates that the Remote Management users are limitted to the execute function. If a different group is used, such as `WinRMRemoteWMIUsers_`, then the RootSDDL line should change accordingly.

8. After this permission is set, run the `Restart-Service WinRM` command to restart the WinRM service.

HTTP Connection

The HTTP listener is usually enabled by default if the WinRM is currently enabled; however, the `AllowUnencrypted` parameter must be set to `True` in order to enable a connection over the HTTP protocol.

1. To enable an HTTP connection, run:

   ```
   WinRM set winrm/config/service '@{AllowUnencrypted="true"}'
   ```

2. Restart the WinRM service again.
3. If a firewall is being used, check that there is a rule to allow inbound connections from the SmartConnector on port 5985 and protocol TCP.

   This rule should have been created when the WinRM service was enabled, but it can be created manually as well.

HTTPS Connection

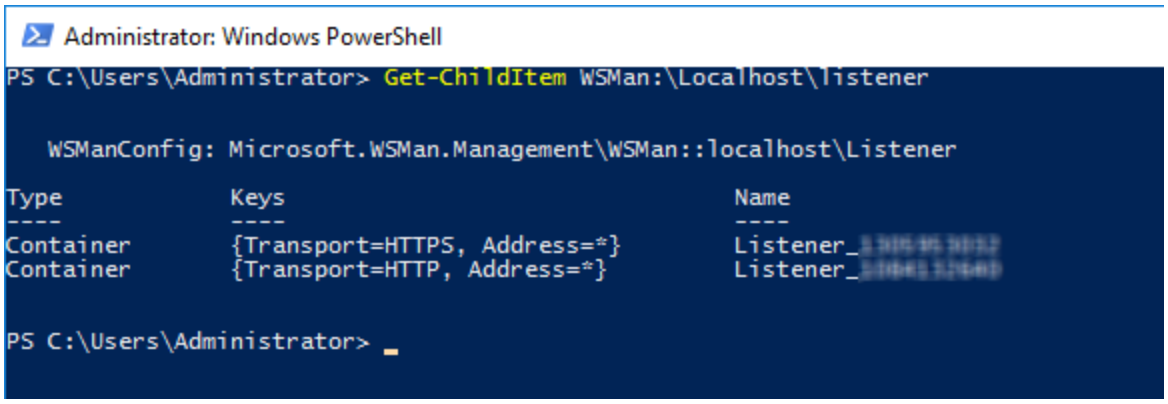
The HTTPS listener is not usually enabled by default. Run the `Get-ChildItem WSMan:\Localhost\listener` command from a PowerShell terminal to check if the HTTPS listener is available or not. The result should contain a line indicating that an HTTPS listener is already existent. For example:

```
Type            Keys                                Name

----            ----                                ----

Container       {Transport=HTTPS, Address=*}        Listener_1305953032
```

The next image shows a machine with both listeners enabled:



If the machine does not have the HTTPS listener enabled then just follow the next steps to create one:

-On a Powershell terminal run the next commands:

- `$cert = New-SelfSignedCertificate -CertstoreLocation Cert:\LocalMachine\My -DnsName <server hostname>`
- `Export-Certificate -Cert $cert -FilePath C:\temp\cert`

  (`C:\temp\` folder must exist before running the command above, cert is a name, not a folder, so, don't create anything inside `C:\temp\`)
- `New-Item -Path WSMan:\LocalHost\Listener -Transport HTTPS -Address * -CertificateThumbPrint $cert.Thumbprint –Force`

Note: Some older Windows systems such as Windows Server 2008 R2 don't support some of the commands used for the self-signed certificate generation shown above, in this scenario please refer to the appendix alternative method to create the HTTPS listener using commands compatible with these older Windows versions.

After these commands are executed, run again the command to check if the new HTTPS listener was created successfully: `Get-ChildItem WSMan:\Localhost\listener`.

Finally, if a firewall is being used then an exception must be created on port TCP 5986 in order to allow inbound requests from the smart connector to this Window host, for example, the next command can be used to create such rule:

```
New-NetFirewallRule -DisplayName 'Windows Remote Management (HTTPS-In)' -Name 'Windows Remote Management (HTTPS-In)' -Profile Any -LocalPort 5986 -Protocol TCP
```

Note: There are other ways to create firewall rules, not just using commands on powershell, the line above is just a functional example on how to do it, just keep in mind to open the correct port (5986) and protocol (TCP) for the correct profile when creating the rule.

After finishing the creation of the HTTPS listener and its firewall rule just restart the WinRM service again and the host is ready to use on the smart connector.

Add Security Certifications When Using SSL for Microsoft Active Directory

If you choose to use SSL for Microsoft Active Directory as the connection protocol, security certificates for both the Windows Domain Controller Service and for the Active Directory Server are required. Installing a valid certificate on a domain controller permits the LDAP service to listen for, and automatically accept, SSL connections for both LDAP and global catalog traffic.

The certificates will be imported to the connector's certificate store during the connector installation process. See **step 3** of the installation procedure for instructions.

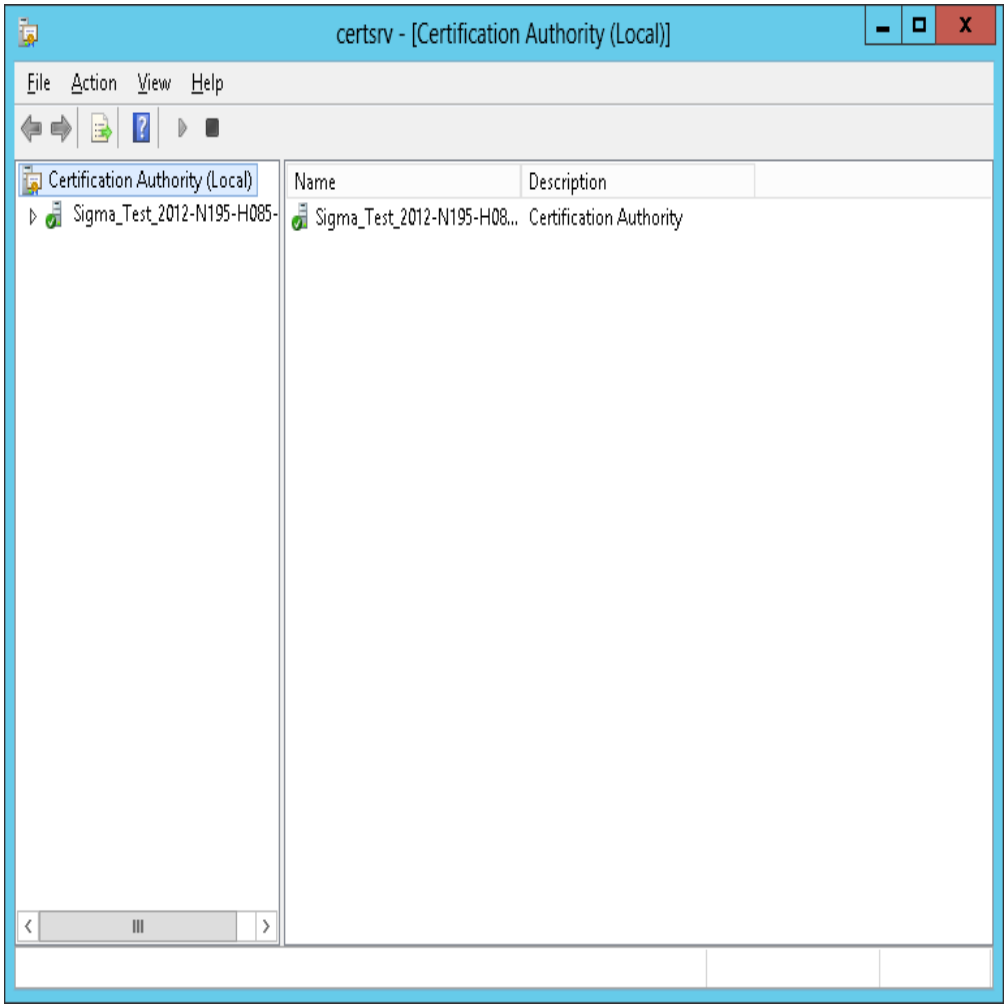
Procedures for Windows 2012 are shown; steps could vary with different Windows versions. For other Windows versions, see Microsoft's documentation for complete information.

Example: Windows Server 2012

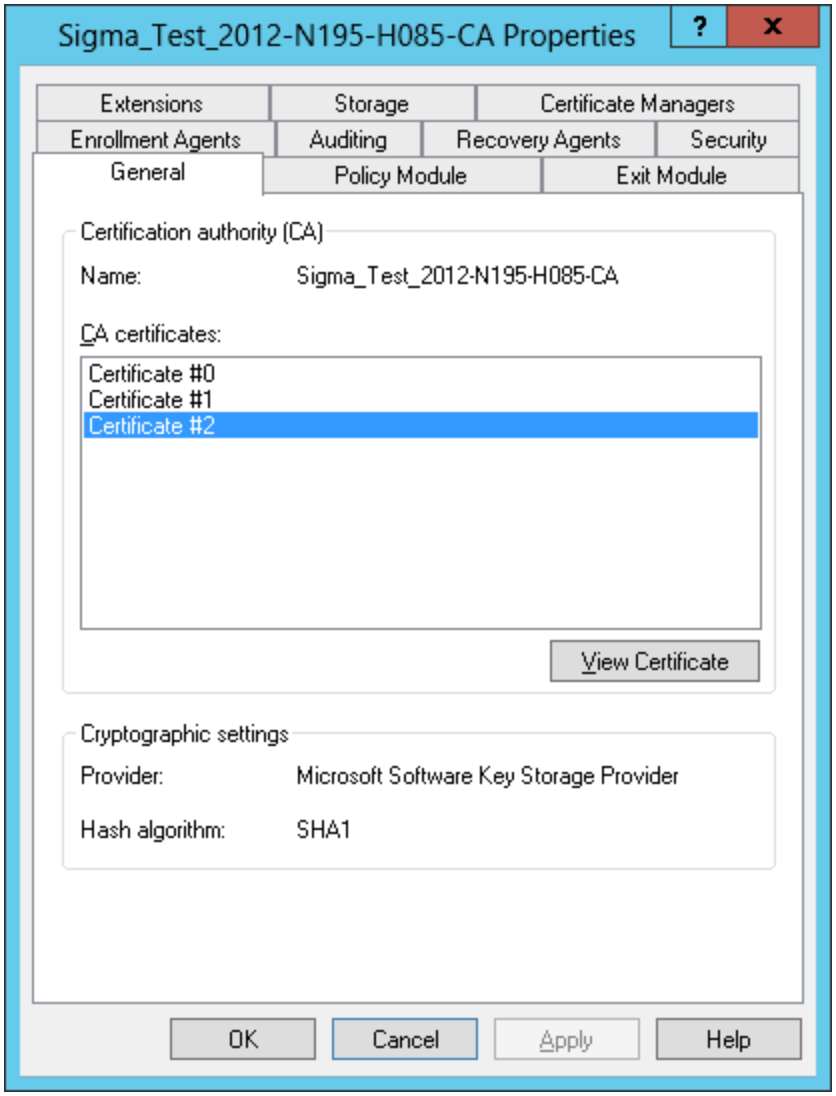
The following steps assume Windows Server 2012 as the operating system

To export the certificates:

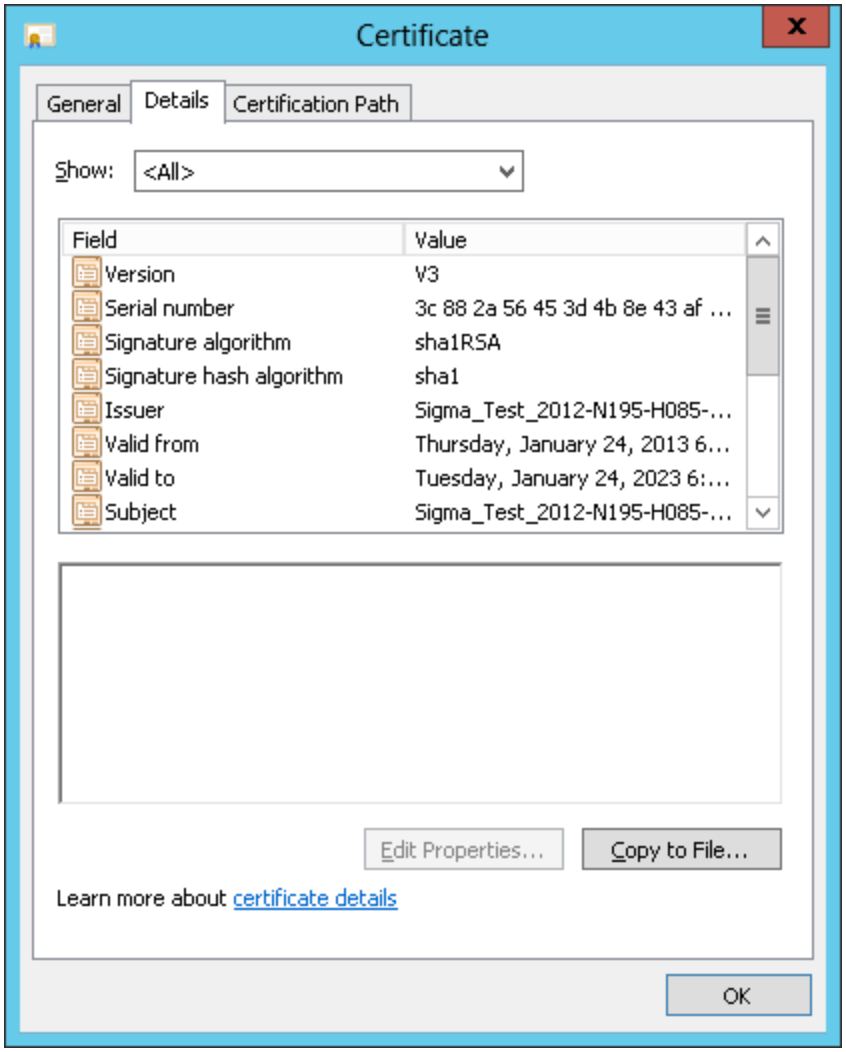
1. From the Windows **Start** menu, select **Administrative Tools**.
2. Select and double-click **Certification Authority**; one or more Domain Certificate Authority servers are shown.



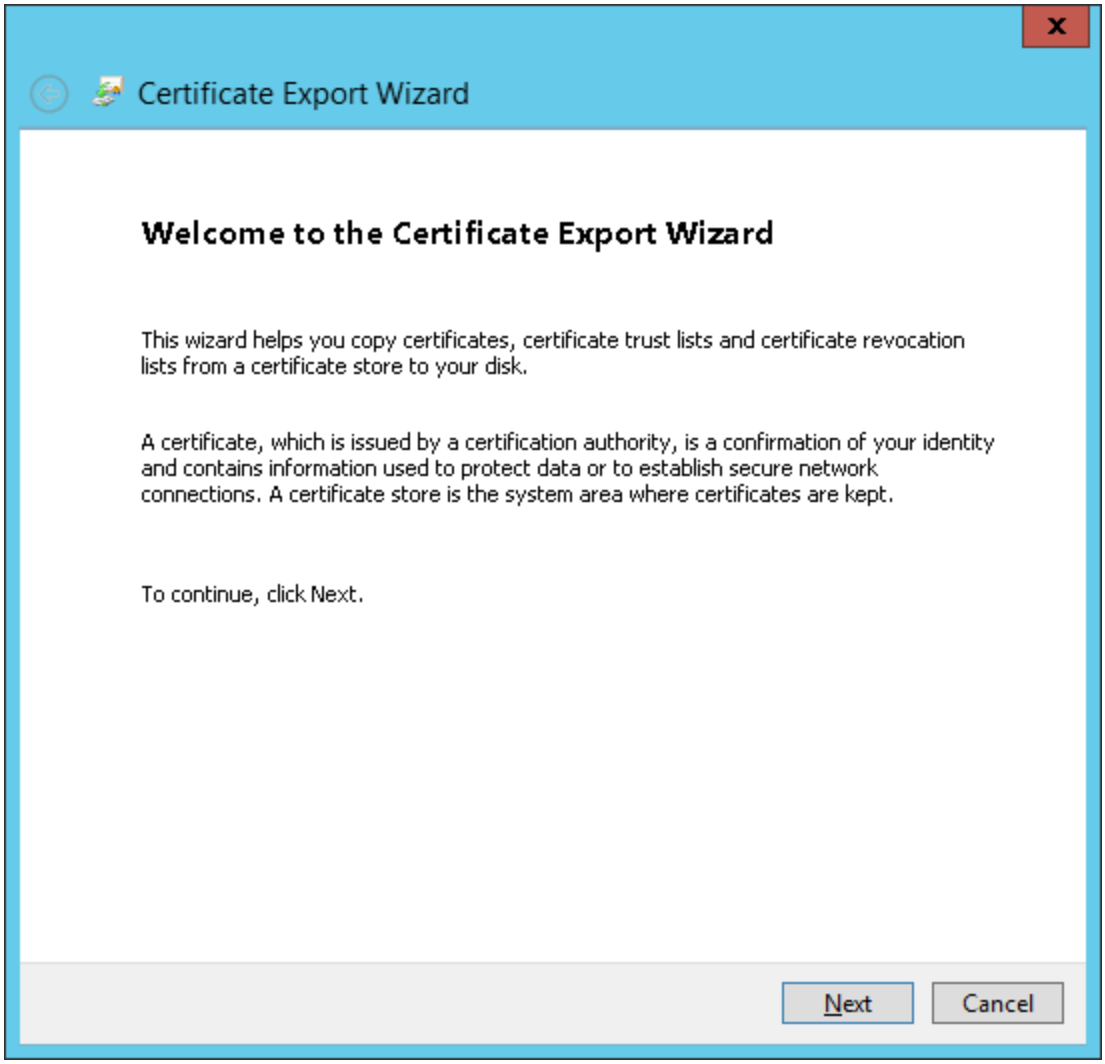
3. Select the Domain Certificate Authority server for the domain to which the Active Directory server belongs, right-click, and select **Properties** to open the **Properties** window.



4. Click **View Certificate**.
5. Click the **Details** tab, and **Copy to File...**



6. Follow the steps in the **Certificate Export Wizard** to complete the export.


Certificate Export Wizard
Welcome to the Certificate Export Wizard
This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.
A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.
To continue, click Next.
Next
Cancel

Chapter 5: Collect Forwarded Events

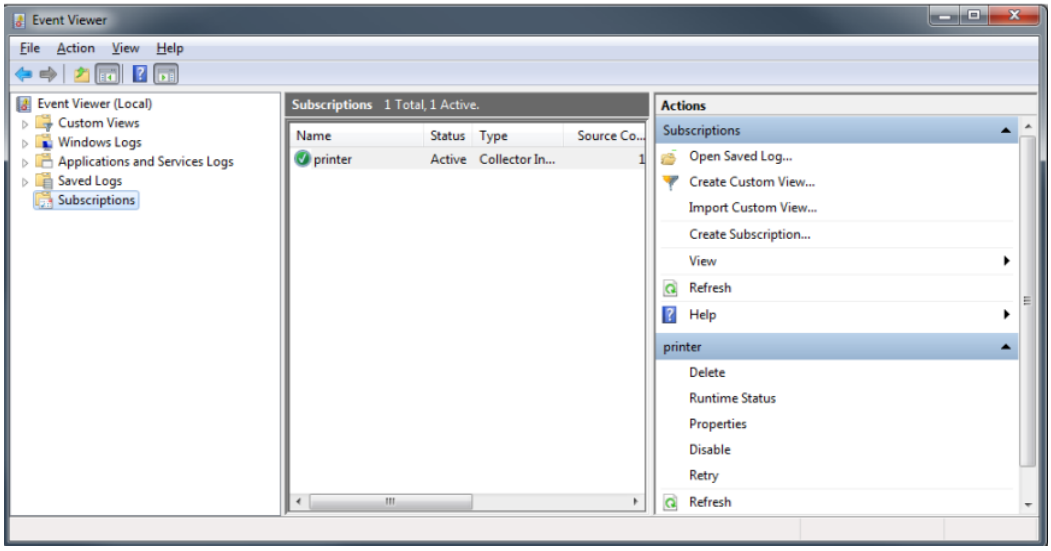
The connector provides a feature to read events forwarded to a Windows Event Collector host. Windows Event Collection is a Microsoft capability that lets a Windows host collect events from multiple sources. Collecting forwarded events is somewhat different than the traditional event collection because the events are from multiple sources.

With Microsoft Windows Event Collector (WEC), you can subscribe to receive and store events on a local computer (event collector) that are forwarded from any number of remote computers (event sources). Before using this feature, read about Windows Event Collector to understand how it works in the Microsoft Windows documentation.

Note: When configuring Windows Event Collection (WEC), Microsoft by default adds to every forwarded event a RenderingInfo section that is a textual description of an event. Having this extra section introduces negative impacts on the resource usage of the WEC machine as well as the performance of the connector. Therefore, Micro Focus advises that you disable the RenderingInfo section. To do so, run the following command from the Windows command console:

```
wecutil ss <subscription-name> /cf:events
```

where subscription-name is the WEC configuration created for event forwarding. This can be found in the **Event Viewer > Subscriptions** folder (see below).



Event Collector for Windows Event Forwarding

You can forward events from a source host to any log type on the collector machine to which the connector would normally have access.

Note: Security events cannot be forwarded to the Security event log on a collector machine, but can be forwarded to other log types.

Source Hosts Windows OS Version

When the connector is configured with the log that has forwarded events, the Windows OS version of the event source host is not populated automatically in the normalized events. To have this value populated, the Windows OS version should be provided as a source host file or the Active Directory should be configured. If the Windows OS version is available from the source host file as well as Active Directory, the value from Active Directory takes precedence.

Active Directory as Source for OS Version

When this selection is chosen during connector configuration, the connector pulls the host information (host name and version) from the configured Active Directory to identify the event source host Windows version information. Newly discovered hosts are added to the lookup automatically without reconfiguring the connector itself.

Active Directory information is checked upon connector startup and every 24 hours (86400000 milliseconds). To change the time setting, locate the `agent.properties` file in `$ARCSIGHT_HOME/current/agent` and set the **hostbrowsingthreadsleeptime** parameter to the number of milliseconds between host browsing queries.) This value should be greater than 0; if the value is set to 0, it will not perform periodic host browsing.

For the connector to be able to browse the Active Directory to retrieve source host Windows version information, it should be placed within the same forest as the Active Directory.

File as Source for OS Version

When this selection is chosen during connector configuration, create a source host file in .csv format that contains the host name and Windows OS version and upload this file during the connector installation/configuration process (the WEF Source Hosts File Name in step 9).

Note: The host file, which is imported to or exported from the host table during installation, and the source host file specified in the **WEF Source Hosts File Name** field are two different entities. The source host file contains only the host name and version information to populate the version in the device version field.

When creating a source host file, make sure to specify the FQDN registered with Active Directory, as the connector finds the version information using the computer name in the event. An example of the source host file could be:

```
hostsa.domaina.com,Windows 7
hostsb.domainb.com,Windows 8
hostsc.domainb.com,Windows Server 2012
Hostsd.domaind.com,Windows Server 2016
```

The valid versions descriptions (case sensitive) that can be used in source hosts files are:

```
Windows Vista
Windows Server 2008
Windows Server 2008 R2
Windows Server 2012
Windows Server 2012 R2
Windows Server 2016
Windows 7
Windows 8
Windows 10
```

Note: OS version information is optional; events may still be parsed in a majority of cases.

Once configured, the OS version is loaded from the source host file when the connector is running on its first run, and is reloaded on the next startup of the connector when the source host file has a timestamp different from the one loaded from the last file processed.

The device version will not be populated in the normalized events.

Chapter 6: Install the SmartConnector

Before you install any SmartConnectors, make sure that the ArcSight products with which the connectors will communicate have already been installed correctly (such as ArcSight ESM or ArcSight Logger). This configuration guide takes you through the installation process with **ArcSight Manager** (**encrypted**) as the destination.

For complete product information for ArcSight products with which the connectors will communicate, see the *Administrator's Guide* and the *Installation and Configuration* guide for your ArcSight product before installing a new SmartConnector.

If you are adding the connector to the ArcSight Management Center (ArcMC), see the *ArcSight Management Center Administrator's Guide* for instructions, and start the installation procedure at "Set Global Parameters (optional)" or "Select Connector and Add Parameter Information".

Required Items

The following items are required when installing this SmartConnector :

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

Installation Notes

- Install this SmartConnector only on 64-bit Linux platforms.

  See "Operating Systems Support for Event Collection."
- It is not possible to upgrade from the Microsoft Windows Event Log -- Unified, or from the Microsoft Windows Event Log - Native connector to the Microsoft Windows Event Log connector.
- If you use Forwarded Event Collection, the full computer name and OS version of source hosts must be available for use either through Active Directory or a source hosts file in csv format.

Install Core Software

1. Download the SmartConnector 64-bit executable for your operating system from the Micro Focus Software Support site.

   `https://softwaresupport.softwaregrp.com/`
2. Start the SmartConnector Installer by running the executable.

   Follow the installation wizard through the following folder selection tasks:

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

3. When the SmartConnector core component software is installed, the Connector Setup window appears, prompting you to add a connector. Set global parameters is the alternate option.

Set Global Parameters (optional)

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

Global Parameter	Setting
Remote Management	Set to **Enabled** to enable remote management from ArcSight Management Center. When queried by the remote management device, the values you specify here for enabling remote management and the port number will be used. Initially, this value is set to **Disabled**.
Remote Management Listener Port	The remote management device will listen to the port specified in this field. The default port number is 9001.
Preferred IP Version	When both **IPv4** and **IPv6** IP addresses are available for the local host (the machine on which the connector is installed), you can choose which version is preferred. Otherwise, you will see only one selection. The initial setting is **IPv4**.

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

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.

Global Parameter	Setting
Format Preserving Identity	The Micro Focus SecureData client software allows client applications to protect and access data based on key names. This key name is referred to as the identity. Enter the user identity configured for Micro Focus SecureData.
Format Preserving Secret	Enter the secret configured for Micro Focus SecureData to use for 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 "Add a Connector."

Use SSL for Connection (optional)

If you are using SSL for connector connection, follow these steps; otherwise, continue with **step 4**.

To import the certificates to the connector's certificate store, click **Cancel** to exit the wizard:

1. From `$ARCSIGHT_HOME\current\bin`, execute the **keytool** application to import the two certificates (see "Add Security Certifications when Using SSL" earlier in this guide).

   ```
   arcsight agent keytoolgui
   ```

   The graphical interface asks you to open a keystore
2. Select `jre/lib/security/cacerts`, then select **import cert** to import your certificate. Verify that the correct certificate has been imported.
3. When prompted **Trust this certificate?**, click **Yes**.

   Repeat this process for the second certificate.
4. Save the keystore.
5. Verify the imported certificates by entering this command from `$ARCSIGHT_HOME\current\bin`:

   ```
   arcsight agent keytool -list -store clientcerts
   ```

   The new certificates are listed.
6. Return to the configuration wizard by entering the following command from `$ARCSIGHT_HOME\current\bin`:

   ```
   runagentsetup
   ```

Add a Connector

1. Select **Add a Connector** and click **Next**.
   The Configuration Wizard displays a list of available SmartConnectors you can configure.
2. Select **Microsoft Windows Event Log** and click **Next**.

Chapter 7: Configure the Connector

Select **Microsoft Windows Event Log** to display the configuration window where you can specify:

- Source hosts for all forwarded events
- Parameters to add hosts for event collection

Source Hosts for All Forwarded Events

If you will be using the connector to collect from forwarded (or WEF) logs, the connector needs to know the Windows OS version for the hosts from which you want to collect events. You can supply a `.csv` file containing this information, or you can let the connector access Active Directory for the host OS version information. Select the appropriate source.

When you select **Use file for OS version**, a window is displayed for you to supply the name of the source hosts file. This is the same window displayed when you select **ForwardedEvents log** in the **Select Logs** section of the initial configuration window. See "WEF Source Hosts File Name."

When you select **Use Active Directory for OS version**, a window is displayed for you to enter your domain credentials and Active Directory parameter information. See "Domain Credentials" and "Active Directory Parameters."

When you select **Do not use any source for Windows OS version**, an Active Directory query or a CSV file to list all hosts involved in events forwarding along with their Windows OS version is not required. No Windows OS version will be displayed in the event headers from the forwarding host.

Parameters to Add Hosts for Event Collection

You can add hosts for event collection using common domain credentials, using Active Directory, or by entering host information manually.

The default domain name, user, and password are used if **Use Active Directory** is checked and values provided in the Active Directory configuration window. Otherwise, specify user name, password, and domain name. When using forwarded event collection, specify only the Event Collector hosts.

If you select **Use common domain credentials**, a window appears where you enter your domain credentials. See "Domain Credentials."

Domain Credentials

Enter the parameter information in the ArcSight Configure dialog and then click **Next**.

Note: A Domain User Name and Domain User Password is not required if you are performing local event collection.

Parameter	Description
Domain Name	Enter the name of the domain to which the host belongs. Work group hosts and stand-alone hosts can be added manually on the table parameters entry window.
Domain User Name	Enter the name of the user account with adequate privileges to collect Windows events from the target host. It is assumed that the AD server is located on the domain server and can be accessed with the domain user and password.
Domain User Password	Enter the password for the user specified in the **Domain User Name** field.

Active Directory Parameters

If you select **Use Active Directory**, a window appears where you specify your domain credentials and Active Directory parameters. This is the same window displayed when you select **Use Active Directory for OS version** in the "Source Hosts for All Forwarded Events" section of the initial configuration window.

For a description of **Domain Name**, **Domain User Name**, and **Domain User Password**, see "Domain Credentials."

Enter the parameter information and click **Next**.

Note:

- If the hosts Domain parameters are the same as Active Directory, then you do not have to enter both. The information will be taken from the Active Directory Domain and credentials.
- If GUID translation is enabled, then the Active Directory Domain and credentials are used. You must provide the complete domain name, including any qualifiers, such as `.com`.

Parameter	Description
Active Directory Domain	Enter the name of the Active Directory domain to which the host belongs.
Active Directory User Name	Enter the name of the user account with adequate privileges to collect Windows events from the target host. It is assumed that the AD server is located on the domain server and can be accessed with the domain user and password.
Active Directory User Password	Enter the password for the user specified in the **Active Directory User Name** field.
Active Directory Server	Enter the Active Directory Host Name or IP address required for authentication to the Microsoft Active Directory for the host browsing feature.

Parameter	Description
Active Directory Filter	Enter the Active Directory Filter required for automatic host browsing to filter hosts by name, operating system, and creation time. The query can contain attributes for Common Names (cn), Operating System (`operatingsystem`) and Creation Time (`whencreated`) in 'YYMMDDHHmmSS' format, where YY=Last two digits of the year, `MM`=Month, `DD`=Date, `HH`=Hours, `mm`=Minutes, `SS`=Seconds in 24-hour format. The query can also contain wildcard characters (*) to match the attributes to different values. **Active Directory Filter examples** To create hosts after and inclusive of a particular time point, set filter to:`(&(cn=*)(operatingsystem=*)(whencreated>=YYMMDDHHmmSSZ))` To create hosts between and inclusive of two time points, set filter to: `(&(cn=*)(operatingsystem=*)(whencreated>=YYMMDDHHmmSS)(whencreated<=YYMMDDHHmmSS))`
Active Directory Protocol	Select whether the protocol to be used is **non_ssl** (the default value) or SSL. For SSL protocol, be sure to import the Active Directory security certificate to the connector before starting the connector.
Use Active Directory host results for	For WEF Only: If you selected "**Use Active Directory for OS Version**" on the initial configuration window, the list of hosts retrieved from Active Directory is used to determine the Windows OS version for the WEF source hosts. When **For WEF Only** is selected, the result of the query will not populate the table of hosts on the table parameter entry window. For initial installation, **Merge Hosts and Replace Hosts** act the same because only the local host is present and preserved. If you selected **Use Active Directory** on the initial configuration screen under **Parameters to add hosts** for event collection, or you are modifying parameters to add hosts, the following applies. When **Merge Hosts** is selected, Active Directory is used to retrieve the hosts for collection (and can also be used for Windows Event Forwarding if WEC servers are present and **Use file for OS** is not selected on the initial configuration screen). The original host is not replaced and all other preconfigured hosts are preserved. Hosts are added from the list retrieved from Active Directory with Security events selected by default. If duplicates are found, the existing host entry is not overwritten. When **Replace Hosts** is chosen, Active Directory is used to retrieve the hosts for collection (and can also be used for Windows Event Forwarding when WEC servers are present and Use file for OS is not selected on the initial configuration screen). The local host is not replaced, but all other hosts preconfigured are replaced with those retrieved from Active Directory, with Security events selected by default.

Enabling FIPS Mode

Prerequisites

1. Import certificates.

   The server certificate check-up is always enabled.

   Note: FIPS Mode can only be enabled if the HTTPS Listener port (5986) is selected.

Procedure

1. Import the certificate into the `bcfips_ks`, Connector truststore (the client).
2. Use the command. `keytool -import -file /PATH_TO_CERT/cert -keystore $ARCSIGHT_HOME/user/agent/fips/bcfips_ks -storetype BCFKS -storepass 'changeit' -provider org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath $ARCSIGHT_HOME/lib/agent/fips/bc-fips-1.0.0.jar -"alias" “wiscBCCert” -noprompt -J-Djava.security.egd=file:/dev/./urandom –v.`
3. Once the certificates are imported, continue with the connector configuration.

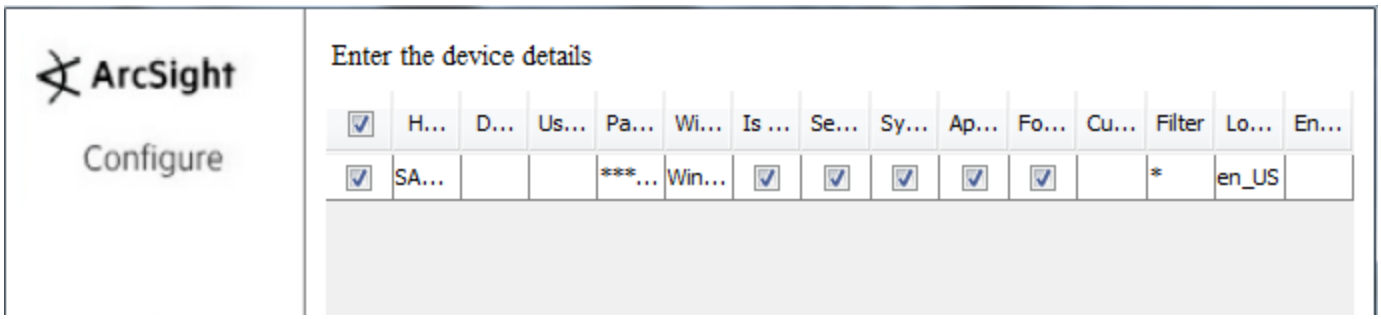
Configure Multiple Host Parameters

If you are adding hosts, a table parameter entry window is displayed (see example on following page).

Selections from the initial parameter entry window for the local host are reflected in the first row of the table. Local host is not supported; therefore, you can either uncheck the row, or simply ignore the row, but do not update the row.

For additional hosts, domain credentials and Windows Version information supplied in a file or through Active Directory are displayed, with only **Security** log selected. You can select other options and provide custom log and filter information for each additional host manually.

If you have added hosts for which you decide not to collect events, you can use the check box in the leftmost column to deselect rows in the table.



The parameters for each host are given in full along with descriptions in the following table. Select options and provide custom log and filter information for each additional host manually.

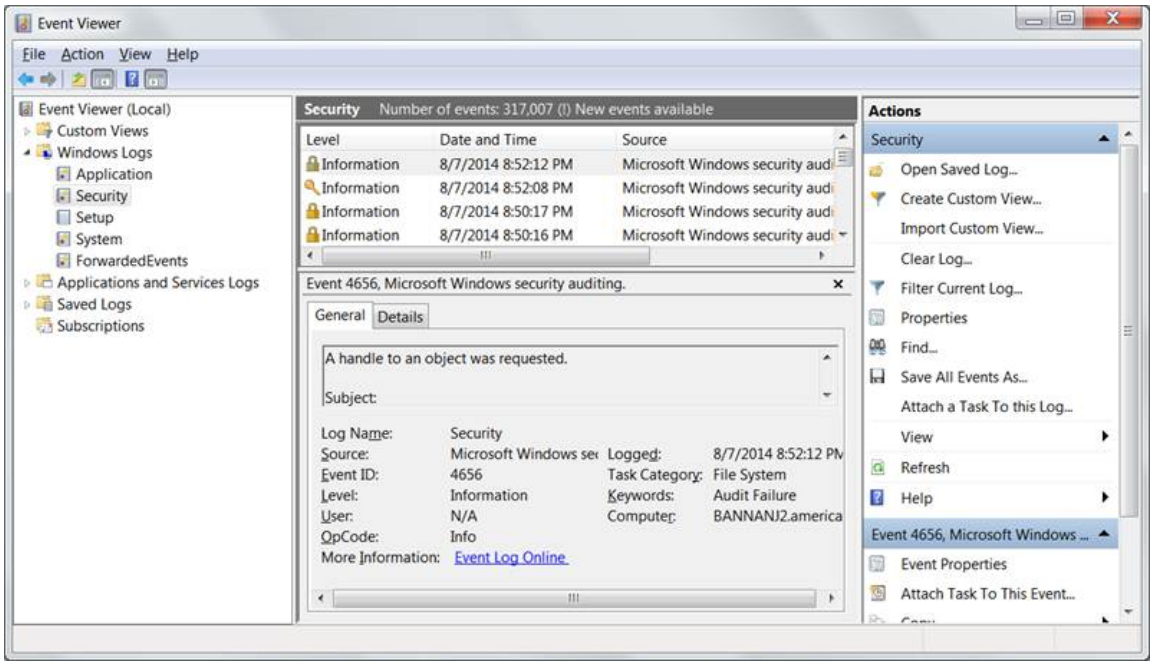
Parameter	Description
Host Name	Host name or IP address of the target Windows host.
Domain Name	Name of the domain to which the host belongs. If you are using a Domain User account for a target host or using Active Directory, fill in the Domain Name field. This must be a name, not an IP address, for the OS version to be resolved.
User Name	Name of the user account with adequate privileges to collect Windows events from the target host. This will be the user name only, without the domain.
Password	Password for the user specified in **User Name**.
Windows Version	Select the Microsoft Operating System version this host is running.
Is WEC	If you selected **Indicates that this is a WEC server** on the initial configuration page, this selection is already checked for the local host.
Security	Select for security events to be collected from this host. This log is automatically selected for all hosts.
System	Select for system events to be collected from this host.
Application	Select for application events to be collected from the **Common Application Event Log** of this host.
ForwardedEvents	Select for events to be collected from the **ForwardedEvents** log of this host.
Custom Event Logs	Specify the custom application log names, separated by a comma (such as "`Exchange Auditing, Directory Service`"). For Windows Event Collector servers, use **HardwareEvents**. See ""Specify Custom Log Names" on page 41" for more information.
Filter	This is a filter you can get from the Microsoft event viewer when you want to collect particular events. You can copy the filter text to this field. For more information, see "Configure a Filter."
Locale	United States English default, **en_US**, is currently supported.
Encoding	The default: `UTF-8`, is currently supported.

After entering the parameter information, click **Next**.

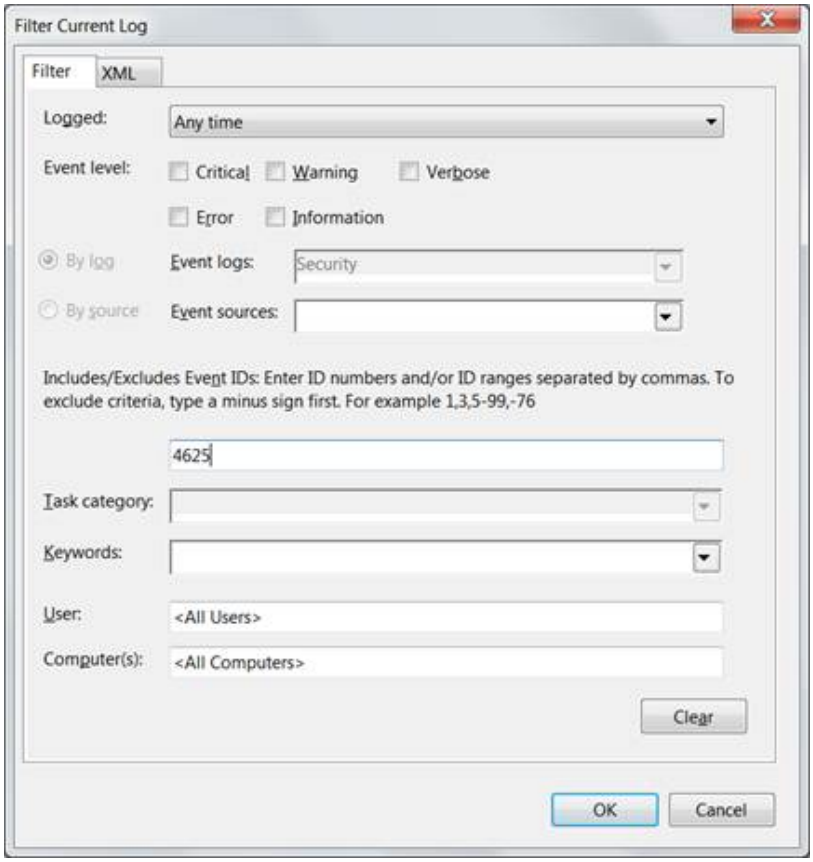
Configure a Filter

To configure a filter, first launch the event viewer and select the event log that needs the filter setting.

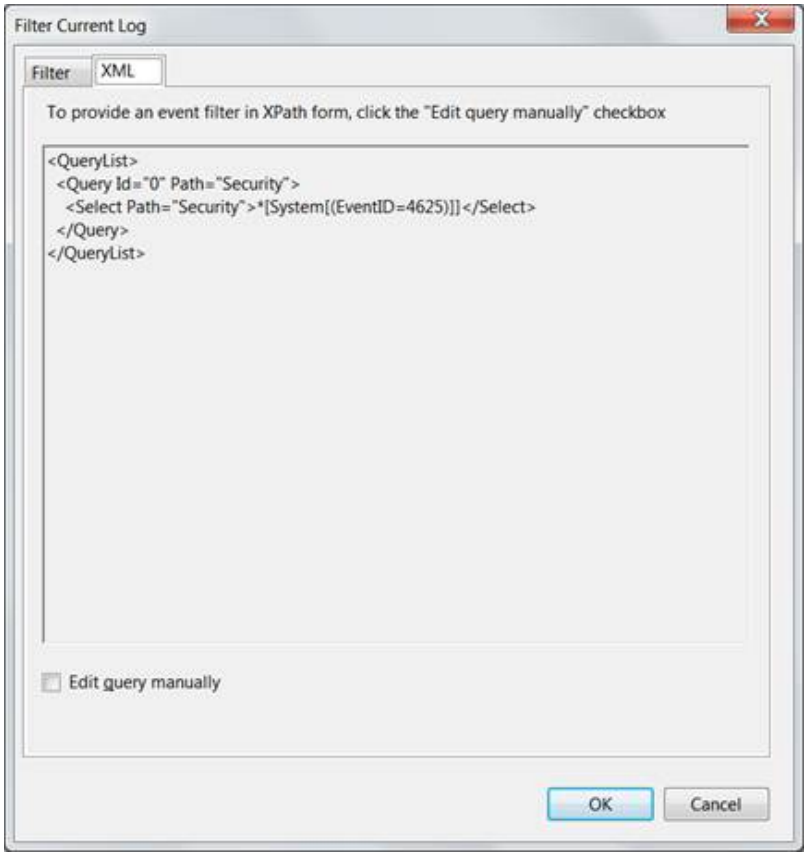
1. Click **Filter current log** to set the filter.



   For example, to collect the logon failure events whose Event ID is 4625, enter the Event ID number as shown in the following figure.



2. Click the **XML** tab. The query is displayed in XML.



The expression that appears between `<Select>` and `</Select>` is the value that can be entered in the filter. Here it writes `*[System[(EventID=4625)]]`. This can be copied to the **Filter** column in the host table parameter for the desired event log.

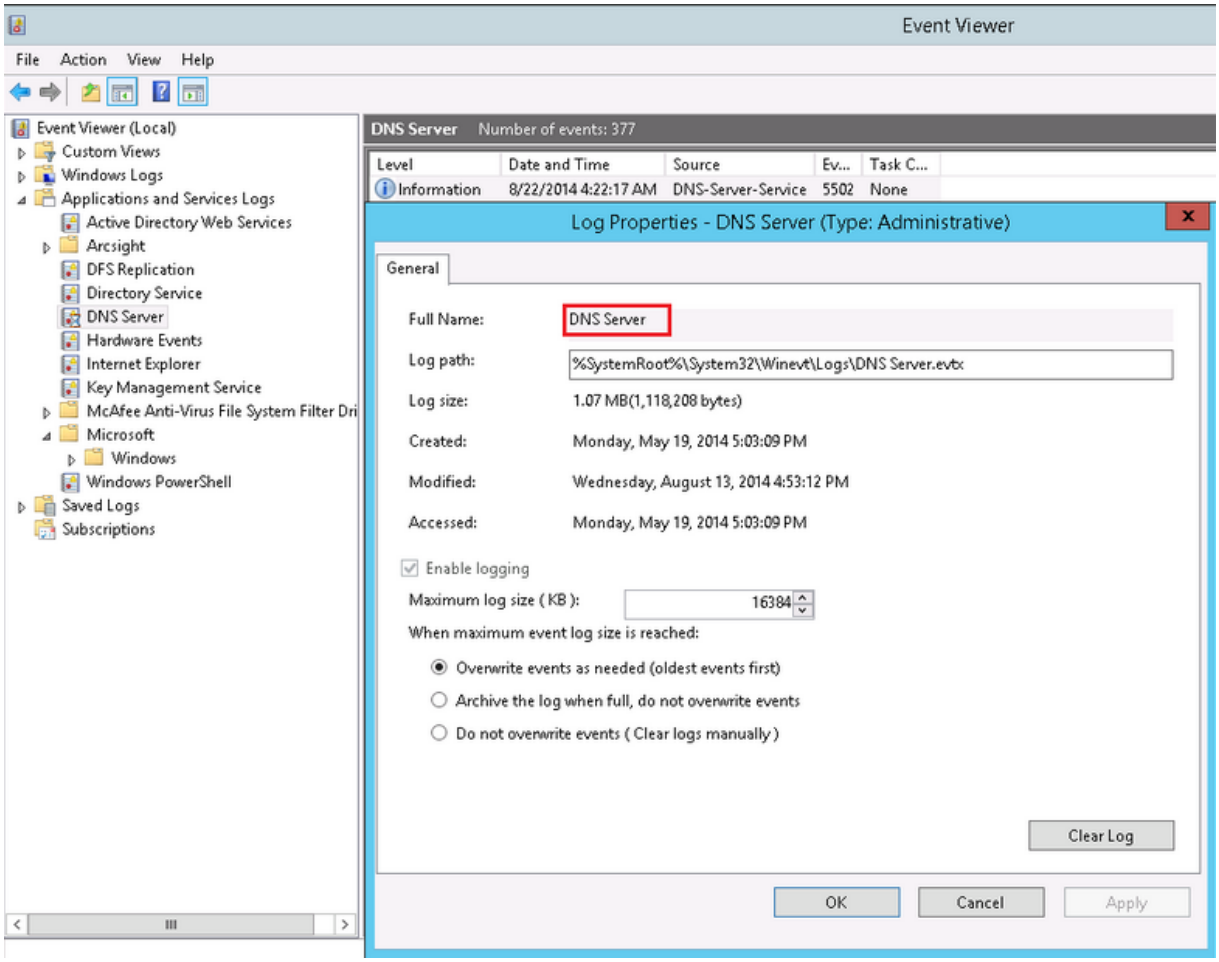
Note: In certain cases, the text cannot be directly copied to the Filter column in the UI wizard. If the filter text contains “gt;”, “lt;”, “gt;=” or “lt;=” , then you must replace it with “>”,“<”, “>=” or “<=” respectively.

Specify Custom Log Names

In the Windows Host parameters window, a column for the **Custom Log Names** parameter lets you specify names of custom event logs. Applications also can generate events for a custom application event log, such as DNS Server, Directory Service, Exchange Auditing, and so on. (Parsing support for only the event header is supported for application events.)

For example, specify `Directory Service` for Active Directory and `Exchange Auditing` for Microsoft Exchange Audit. For Microsoft Windows Print Service Admin log, use `Microsoft-Windows-PrintService/Admin`.

To identify the Custom Event Log Name, select the **Custom Application Event Log** in the Microsoft Windows **Event Viewer**. The log name can be found from the properties of the event log in the **Full Name** field, as shown in the following figure.



For more information about setting this parameter, see "Advanced Configuration Parameters per Host."

WEF Source Hosts File Name

When you have selected **ForwardedEvents log** from the **Select logs for event collection from local host** section or **Use file for OS version** from the **Source hosts for all forwarded events** section (and have not selected Use Active Directory), a window appears where you enter the name of the file containing the source host information. This window is also displayed if you have selected **Is WEC** for any hosts in the table parameter window.

Enter the source hosts file name and click **Next**.

Configuration Summary

When you have completed configuration, click **Next** and a window summarizing your selections is displayed. Destination configuration begins with the next window. See "Select a Destination."

Chapter 8: Select a Destination

This section documents forwarding of events to ArcSightESM.

1. Make sure **ArcSight Manager** (encrypted) is selected and click **Next**. For information about other, see the *ArcSight SmartConnector User Guide* as well as the Administrator's Guide for your ArcSight product.
2. Enter the **Manager Host Name**, **Manager Port**, and a valid ArcSight **User Name** and **Password**. This is the same user name and password you created during the ArcSight Manager installation. For a complete description of the destination parameters, see the *ArcSight SmartConnector User Guide*. Click **Next**.
3. Enter a **Name** for the SmartConnector and, optionally, **Location**, **Device Location**, and **Comment** to identify the connector's use in your environment. Click **Next**; the connector starts the registration process.
4. The certificate import window for the ESM 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.
5. The certificate is imported and the **Add Connector Summary** window is displayed.

Chapter 9: Finalize Installation and Configuration

To finalize the SmartConnector installation and configuration process, follow these steps:

1. Review the **Add Connector Summary** and click **Next**. If the summary is incorrect, click **Previous** to make changes.
2. The wizard now prompts you to choose whether you want to run the SmartConnector as a stand-alone process or as a service. If you chose to run the connector as a service, with **Install as a service** selected, click **Next**. The wizard prompts you to define service parameters. Enter values for **Service Internal Name** and **Service Display Name** and select **Yes** or **No** for **Start the service automatically**. The **Install Service Summary** window is displayed when you click **Next**. If you choose to run the connector as a stand-alone process, you will not be asked to supply service parameters and can skip the next step. For more information about running the connector as a service or daemon, see the *SmartConnector User Guide*.

   Note: ArcSight recommends installing this connector as a service. If the connector is started as a standalone application and CTRL+C is used to shut down the connector, the connector's "WINC agent" process may not persist in the SID cache and exit prematurely.

3. Enter the service parameters and click **Next**. The **Install Service Summary** window is displayed.
4. Click **Next**.
5. To complete the installation, choose **Exit** and click **Next**.

   For some SmartConnectors, a system restart is required before the configuration settings you made take effect. If a **System Restart** window is displayed, read the information and restart the system.

   Note: Save any work on your computer or desktop and shut down any other running applications (including the ArcSight Console, if it is running), then shut down the system.

   Continue with "Run the SmartConnector ." For connector upgrade or uninstall instructions, see the *SmartConnector User Guide*.

Chapter 10: Run the SmartConnector

SmartConnectors can be installed and run in standalone mode or on Windows platforms as a Windows service. The SmartConnectors also can be run using shortcuts and optional **Start** menu entries.

If installed standalone, the connector must be started manually, and is not automatically active when a host is restarted. If installed as a service, the connector runs automatically when the host is restarted. For information about connectors running as services, see the *ArcSightSmartConnector User Guide*.

For connectors installed standalone, 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 SmartConnector logs, read the `agent.log` and `wincagent.log` files located at `$ARCSIGHT_HOME\current\logs\`; to stop all SmartConnectors, enter CTRL+C in the command window.

Chapter 11: Modify Configuration Parameters

To modify configuration parameters, go to `$ARCSIGHT_HOME\current\bin` and double-click `runagentsetup.bat`.

1. Select **Modify Connector**. Click **Next**.
2. Select **Modify connector parameters**; click **Next**.

Make your changes and continue with the wizard in the same manner as during initial connector configuration. For descriptions of the parameters, see "Configure the Connector."

Chapter 12: Create Custom Parsers for System and Application Events

The SmartConnector provides complete parsing of both the Windows event header and event description for all security events and some system events, as specified in this guide.

For all system and application events, the connector provides complete parsing of the Windows event header. Also, the connector provides a framework for creating and deploying your own parsers to parse the event description. Such a parser can parse events specific to a Channel and ProviderName.

- When collecting events from system event logs (such as NTServicePack, Service Control Manager, WINS), select **System** for **Windows Log type**.
- When collecting events from application event logs (such as Microsoft Forefront Protection 2010 for Exchange, Microsoft SQL Server Audit), select **Application** for **Windows Log type**.

Note: Custom Parsers or overrides you create are customizations. These are not certified for use through the ArcSight Quality Assurance Life Cycle of Testing. These are to be developed, tested, and maintained by the creator of the Custom Parser or override.

Before Creating a Parser

Follow these steps prior to creating a parser:

1. Generate the system or application events of interest.
2. Configure the connector to collect the system or application events and preserve the raw events.
3. Run the connector to collect the system or application events and to generate the ArcSight raw events. The raw events will contain key-value pairs in JSON format. Using these generated raw events, see "Create and Deploy Your Own Parser" to map the values of these keys to the ArcSight event schema fields by creating a parser file.

   Note: Not all raw events will have key-value pairs in the event body. Such events do not require that you create a parser to map anything to the ArcSight event schema fields. But you can still choose to create a parser to map the event name or description for such events.

Create and Deploy Your Own Parser

To create and deploy your own parser:

1. Navigate to the directory location for deploying the parser file:

   ```
   $ARCSIGHT_HOME\user\agent\fcp\winc
   ```

2. Identify the Channel for the events that need to be parsed (for example: System, Application, Directory Service, DNS Server, Key Management Service, and so on).
3. Identify the provider name of the events that need to be parsed, since events collected from a single channel can be generated by multiple provider names. For example, events collected from `Channel: System` can be generated by `ProviderName: Service Control Manager`, `WINS`, and so on.
4. Identify the `SectionName` of the event body that needs to be parsed, such as `EventData`, `UserData`, and so on.
   a. To parse the `EventData` section of the event body, create a key value parser file with the following naming convention, in the directory location identified in **Step 1**.

      ```
      \{Normalized Channel}\{Normalized ProviderName}.sdkkeyvaluefilereader.
      properties
      ```

      For example, the key-value parser file name for:

      - Channel: `Security`
      - ProviderName: `Microsoft Windows Event Log`
      - SectionName: `EventData`

      will be:

      ```
      \security\microsoft_windows_eventlog.sdkkeyvaluefilereader.properties
      ```

   b. To parse the other sections of the event body, such as `UserData`, create a JSON parser file with the following naming convention, in the directory location identified in **Step 1**.

      ```
      \{Normalized Channel}\{Normalized ProviderName}.{Normalized
      SectionName}. jsonparser.properties
      ```

      For example, the key-value parser file name for:

      - Channel: `Security`
      - ProviderName: `Microsoft Windows Event Log`
      - SectionName: `UserData`

      will be:

      ```
      \security\microsoft_windows_eventlog.userdata.jsonparser.properties
      ```

      Note: Normalize the `Channel`, `ProviderName`, and `SectionName` values by changing all letters to lower case, and then replacing each character that is not a letter or digit (including special characters and spaces) with an underscore character (_). Do not normalize the `Locale` and `Encoding` values.

5. Create the mappings in these parsers as per your requirements by using conditional mappings based upon the ArcSight `externalId` field, which is already mapped to the Windows Event ID.

Because the connector already maps the Windows event header fields to ArcSight event fields as previously mentioned, those mappings need not be re-defined (unless you need to override the mapping values). The only mappings required are for mapping the specific event description.

a. The following event header key-value parser can be used as a reference for:

- Channel: `Security`
- ProviderName: `Microsoft Windows Event Log`
- SectionName: `EventData`

to map the event name fields:

```
key.delimiter=&&
key.value.delimiter==
key.regexp=([^&=]+)

event.deviceVendor=__getVendor("Microsoft")

conditionalmap.count=1
conditionalmap[0].field=event.externalId
conditionalmap[0].mappings.count=2

# The event logging service has shut down.
conditionalmap[0].mappings[0].values=1100
conditionalmap[0].mappings[0].event.flexString1=
conditionalmap[0].mappings[0].event.name=__stringConstant("The event
logging service has shut down.")

# The security log is now full.
conditionalmap[0].mappings[1].values=1104
conditionalmap[0].mappings[1].event.flexString1=
conditionalmap[0].mappings[1].event.name=__stringConstant("The security
log is now full.")
```

Be sure no trailing spaces appear in your file after you copy and paste this example.

b. The following `UserData` section from the sample JSON parser below can be used as a reference for:

- Channel: `Security`
- ProviderName: `Microsoft Windows Event Log`
- SectionName: `UserData`

Sample `UserData` section:

```
{
   "UserData": {
      "LogFileCleared":

         "@xmlns:auto-ns3":
"http://schemas.microsoft.com/win/2004/08/events",
         "@_xmlns_":
http://manifests.microsoft.com/win/2004/08/windows/eventlog",
         "SubjectUserSid": "S-1-5-18",
         "SubjectUserName": "SYSTEM",
         "SubjectDomainName": "NT AUTHORITY",
         "SubjectLogonId": "0x3e7"
         }
      }
}
```

c. The following EventBody JSON parser can be used as a reference for:

- Channel: `Security`
- ProviderName: `Microsoft Windows Event Log`
- SectionName: `UserData`

Sample `EventBody` section:

```
trigger.node.location=/UserData
event.deviceVendor=__getVendor("Microsoft")
token.count=7
token[0].name=SubjectUserSid
token[0].location=LogFileCleared/SubjectUserSid
token[0].type=String

token[1].name=SubjectUserName
token[1].location=LogFileCleared/SubjectUserName
token[1].type=String

token[2].name=SubjectDomainName
token[2].location=LogFileCleared/SubjectDomainName
token[2].type=String

token[3].name=SubjectLogonId
token[3].location=LogFileCleared/SubjectLogonId
token[3].type=String

token[4].name=Reason
token[4].location=AuditEventsDropped/Reason
```

```
token[4].type=String

token[5].name=Channel
token[5].location=AutoBackup/Channel
token[5].type=String

token[6].name=BackupPath
token[6].location=AutoBackup/BackupPath
token[6].type=String

conditionalmap.count=1
conditionalmap[0].field=event.externalId
conditionalmap[0].mappings.count=3

conditionalmap[0].mappings[0].values=1101
conditionalmap[0].mappings[0].event.name=__stringConstant("Audit events
have been dropped by the transport. The real time backup file was
corrupt due to improper shutdown.")
conditionalmap[0].mappings[0].event.deviceCustomNumber3=__safeToLong
(Reason)
conditionalmap[0].mappings[0].event.deviceCustomNumber3Label=__
stringConstant("Reason Code")

conditionalmap[0].mappings[1].values=1102
conditionalmap[0].mappings
[1].event.destinationNtDomain=SubjectDomainName
conditionalmap[0].mappings[1].event.destinationUserName=__extractNTUser
(__oneOf(SubjectUserName,SubjectUserSid))
conditionalmap[0].mappings[1].event.destinationUserId=SubjectLogonId
conditionalmap[0].mappings[1].event.name=__stringConstant("The audit
log was cleared.")

conditionalmap[0].mappings[2].values=1105
conditionalmap[0].mappings[2].event.fileType=Channel
conditionalmap[0].mappings[2].event.fileName=BackupPath
conditionalmap[0].mappings[2].event.name=__stringConstant("Event log
automatic backup")
```

Be sure no trailing spaces appear in your file after you copy and paste this example.

6. Start the connector.

Be sure to check categorization of new events; additional categorization could be required. For information about categorization, see the Technical Note *ArcSight Categorization: A Technical*

Perspective available from the Micro Focus Software Support site. For more information about creating parsers, see the *ArcSight FlexConnector Developer's Guide*, available from the Micro Focus Software Support and Protect 724 sites.

Chapter 13: Additional Configuration

Customize Event Source Mapping

The Windows Event Log application/system event parser loading mechanism relies on the event source for each event and attempts to load a parser with the following name convention:

`<Channel>\<ProviderName>.sdkkeyvaluefilereader.properties`

This convention works in the vast majority of cases but sometimes the parser needs more flexibility. In these cases, you can customize where to find these parsers by redirecting the variables `Channel` and `ProviderName`. For even more flexibility, the input `ProviderName` can be matched against a regular expression to avoid duplicate entries with minimal changes.

Create an Override Map File

1. Navigate to `$ARCSIGHT_HOME/current/user/agent/fcp/winc/core_maps` and create an override map file with the name `customeventsource.map.csv` including the following columns:

   ```
   SourceChannel
   SourceProviderNamePattern
   TargetProviderName
   TargetChannel
   ```

   The `SourceProviderNamePattern` value can be a string or a regular expression.
2. If there is no `winc/coremaps` subdirectory at `$ARCSIGHT_HOME/current/user/agent/fcp`, create one.
3. The last field `TargetChannel` is optional and, if empty, will be understood as the same as `SourceChannel`.

Example of Event Parsing in a Clustered Environment

The default parser filename convention can cause problems in clustered environments, where the same event from different clusters can have different customized provider names. For example, SQL Server application events have the ProviderName `MSSQLSERVER`, resulting in a parser name of `application\mssqlserver.sdkkeyvaluefilereasder.properties`.

In a clustered SQL Server environment, you can customize and configure the provider name for each cluster as `SQLSERVER01`, `SQLSERVER02`, and so forth. However, the connector is expecting a provider name of `MSSQLSERVER`, and without some modifications, parsing will fail for events with customized provider names.

To avoid this outcome, you can map all these different providers into one provider name value using the map file `$ARCSIGHT_HOME/user/agent/fcp/winc/core_maps/customeventsource.map.csv`.

The following are example entries based on the above clustered environment:

```
Application, MSSQLSERVER01, MSSQLSERVER, Application
Application, MSSQLSERVER\d*, MSSQLSERVER, Application
Application, MSSQLSERVER.*, MSSQLSERVER, Application
```

The complete contents of a sample `customeventsource.map.csv` file with two entries may appear as:

```
#SourceChannel, SourceProviderNamePattern, TargetProviderName,
 System,        Service.*,                 service_control_manager,
 Application,   MSSQLSERVER.*,             MSSQLSERVER,
```

Chapter 14: Configure Advanced Options

This section documents some of the advanced configuration parameters available with this connector. The table following the procedure for accessing advanced configuration parameters details the parameters you may choose to adjust, depending upon the needs of your enterprise.

Access Advanced Parameters

After SmartConnector installation, you can edit the `agent.properties` file to modify parameters. This file can be found at `$ARCSIGHT_HOME\current\user\agent`.

Advanced Container Configuration Properties

Specify	Parameter	Default
The connection port used by the connector to connect to the Windows host via WinRM. For HTTPS, use port 5986. For HTTP, use port 5985.	winc.winc-agent.wisc.winrm.port	5986
To enable and/or disable SID translation. Acceptable values are true and false.	winc.winc-agent.wisc.enablesidtranslation	true
The maximum disk size (in Kilobytes) to be used for message persistence by the MQ component.	`mq.persistent.storage.limit`	`409600`
The maximum memory size (in Kilobytes) to be used by the MQ component.	`mq.memory.limit`	`65536`
The frequency to clean up the processed messages from persistent store in milliseconds. The storage needs to be cleaned up in order to receive more messages from `winc-agent`.	`mq.persistent.storage.cleanup.interval`	`10000`

Specify	Parameter	Default
The number of messages, event batches to preload in memory. Received messages from the winc-agent are persisted into the memory store, but it has to be loaded into the memory for processing. Preloading reduces the waiting time for the data loading and helps with performance.	`mq.consumer.prefetch.size`	`80`
The number of events retrieved from the stream of events on every read operation. Recommended to be <512 to prevent any performance issue. Possible values <=812.	`winc.winc-agent.wisc.eventCollection.batchSize`	`512`
Initial time in minutesin sleep mode before next poll, in case, the current poll retrieves zero events. If the application event logs generates a few events in 24 hours. If you are concerned about performance, the polling threads must be in sleep mode for some time to avoid busy polling. The minimun value must be less than zero.	`winc.winc-agent.wisc.eventCollection.noEvents.sleepTime.min.ms`	`10000`
Maximum time in sleep mode in minutes before next poll, in case, the current poll retrieves zero events. If the application event logs generates a few events in 24 hours. If you are concerned about performance, the polling threads must be in sleep mode for some time to avoid busy polling. The minimun value must be less than zero.	`winc.winc-agent.wisc.eventCollection.noEvents.sleepTime.max.ms`	`300000`
Size of the queue that keeps the unprocessed collected events in XML format, and its context info.	`winc.winc-agent.wisc.eventProcessing.queueSize`	`500`

Specify	Parameter	Default
Number of threads that processes the collected events	`winc.winc-agent.wisc.eventProcessing.threadCount`	`5`
SID to username translation capability and viceversa . The SID should be present in the remote host. Note: There may be a slight performance hit when being used.	`winc.winc-agent.wisc.enablesidtranslation`	`True`
Enables SID translation for SIDs without {% and } charts.	`winc.winc-agent.wisc.enablesidtranslationalways`	`True`

Advanced Common Configuration Parameters

Specify	Parameter	Default
Thread count for event parsing threads.	`eventprocessorthreadcount`	`20`
The queue size used to hold the ready to execute event processing task to improve performance. Larger queue length means bigger memory footprint and it does not necessarily help with performance improvement, as a limited number of threads are available for processing.	`Executequeuelength`	`500`
By default the statistics are calculated every 10 minutes and dumped into both the `agent.log` and to the `EventStats` report file in `user/agent/agentdata`. This interval governs how often stats are calculated. Stats include average per last interval for events per second.	`pdastatsinterval`	`600000ms`
Whether to preserve the last ID processed before connector terminated or device went down.	`preservestate`	`true`
Event count before writing the preserve state.	`preservedstatecount`	`100`
Time interval in ms before writing the preserve state.	`preservedstateinterval`	`10000`

Advanced Configuration Parameters per Host

Specify	Parameter	Default
To collect application events from custom application event logs, provide a comma separated list of the custom application event logs.	eventlogtypes	null

Advanced Configuration Parameters for GUID Translation

Specify	Parameter	Default
To enable GUID translation	`enableguidtranslation`	`false`
Size of the cache to store the GUIDs and their translated values	`guidcachesize`	`50000`
Time-to-live in ms for the GUID entries in the caches	`guidcachetimetolive`	`600000`
Interval in milliseconds (ms) at which the SID and GUID entries are to be expired from the caches	`sidguidcacheexpirationthreadsleeptime`	`600000`
Interval in ms at which the SID and GID caches are persisted to disk files. Each domain's SID cache is persisted to a separate disk file. The SID cache for workgroup hosts is persisted to a separate shared disk file.	`sidguidcachepersistencethreadsleeptime`	`600000`

Chapter 15: Log message for resource adjustment

Symptom: While the connector is starting, it logs that the temporary store will be downsized.

```
2015-01-26 15:11:17,668][ERROR]
[default.org.apache.activemq.broker.BrokerService]
[external] Temporary Store limit is 51200 mb, whilst the temporary data
directory: C:\arcsight\SmartConnectors\current\activemq-data\localhost\tmp_
storage only has
47568 mb of usable space - resetting to maximum available 47568 mb.
```

Solution: This message indicates that the system disk space is low. Although this may not cause an immediate impact, check for adequate disk storage to ensure it does not run out while running the connector. To avoid this log message, make sure the system has 50 GB of disk space available.

Appendix A: Setup Scenarios

The following examples describe some typical setup scenarios. See "Configure the Connector" for configuration details.

- Collect Application, Security, and System Logs from Windows Hosts, from One Domain, and Enter the Hosts Manually
- Collect Forwarded Events or Other WEC Logs from Windows Hosts

Collect Application, Security, and System Logs from Remote Hosts, from One Domain, and Enter the Hosts Manually

In this scenario, a table parameter entry window appears.

Click **Add** to add a row to the table and enter your host information. Or, you can click **Import** to import a csv file containing your host information.

Collect Forwarded Events or Other WEC Logs from Windows Hosts

With any of the previous scenarios, to collect Forwarded Events or other Windows Event Collector (WEC) logs from the remote hosts, a configuration window appears where you can specify the name of a csv file containing the source hosts names and Windows OS versions for the hosts after making configuration selections for your hosts on the table parameter entry window.

From the ArcSight Configure dialog, you can specify:

- Source hosts for all forwarded events
- Parameters to add hosts for event collection

Appendix B: Types of Internal Events

The Windows Event Log connector documents the following types of internal events:

- Remote Agent Connected
- Remote Agent Configuration Accepted

Remote Agent Connected

Field	Description
Event Name	'Remote Agent Connected'
Device Event Category	'/Informational'
Agent Severity	'2'
Device Custom String 1 Label	'Collector Host Name'
Device Custom String 1	<Collector Host Name>
Device Custom String 2 Label	'Collector Domain Name'
Device Custom String 2	<Collector Domain Name>
Device Custom String 5 Label	'Collector Operating System Version'
Device Custom String 5	<Collector Operating System Version>

Remote Agent Configuration Accepted

Collector Status for "Remote Agent Configuration Accepted"

Field	Description
Event Name	'Remote Agent Configuration Accepted'
Reason	<SuccessStatus/FailureReason>
Device Event Category	'/Informational' or '/Informational/Warning' depending on the reason
Agent Severity	'2' or '3' depending on the reason
Device Custom String 1 Label	'Collector Host Name'
Device Custom String 1	<Collector Host Name>
Device Custom String 2 Label	'Collector Domain Name'
Device Custom String 2	<Collector Domain Name>

Field	Description
Device Custom String 5 Label	'Collector Operating System Version'
Device Custom String 5	<Collector Operating System Version>
Field	**Description**
Event Name	'Remote Agent Configuration Accepted'
Device Host Name	<DeviceHostName>
Reason	<SuccessStatus/FailureReason>
Device Event Category	'/Informational' or 'Informational/Warning' depending on the reason
Agent Severity	'2' or '3' depending on the reason
Device Custom String 1 Label	'Collector Host Name'
Device Custom String 1	<Collector Host Name>
Device Custom String 2 Label	'Collector Domain Name'
Device Custom String 2	<Collector Domain Name>
Device Custom String 5 Label	'Collector Operating System Version'
Device Custom String 5	<Collector Operating System Version>
Event Name	'Remote Agent Configuration Accepted'
Device Host Name	<DeviceHostName>
Device Custom String 3 Label	'Event Log'
Device Custom String 3	<ConfiguredEventLogName>
Reason	<SuccessStatus/FailureReason>
Device Event Category	'/Informational' or '/Informational/Warning' depending on the reason
Agent Severity	'2' or '3' depending on the reason
Device Custom String 1 Label	'Collector Host Name'
Device Custom String 1	<Collector Host Name>
Device Custom String 2 Label	'Collector Domain Name'
Device Custom String 2	<Collector Domain Name>
Device Custom String 5 Label	'Collector Operating System Version'
Device Custom String 5	<Collector Operating System Version>

Appendix C: Microsoft Windows Event Log Connector and Unified Features Comparison

This topic compares the SmartConnector for Microsoft Windows Event Log to the SmartConnector for Microsoft Windows Event Log - Unified.

The connector is ArcSight's Windows Event Log collection SmartConnector. It uses Microsoft technology and has broad capabilities, but can be installed only on Linux.

The Unified connector is ArcSight's legacy Windows Event Log collection SmartConnector. It is a portable connector that can be installed on both Windows and Unix systems. This is achieved through a Java implementation of the Windows logging technology (JCIFS), which limits the connector to JCIFS technical capabilities.

Windows Event Log and Unified Connector Features

Feature	Connector	Unified Connector
Pre-filtering	Performs pre-filtering on the sending server. This conserves bandwidth and enhance connector performance. For example, if you are interested only in logon failures (such as Event ID 4625), you do not need to get any other event to the connector.	Does not perform pre-filtering.
IPv6 Stack	Able to fully run on the IPv6 stack.	IPv6 stack not supported.
Forwarded Events	Collects from ForwardedEvents log, which is the default when you setup a WEF subscription.	Collects remote logs only from the HardwareEvents event log, in addition to Security/Application/System.
Custom Event Logs	Can read events in any Windows event log, including AppLocker and Windows Defender events. The flex framework makes it easier to create custom parsers	Has limitations in reading Windows event logs, although there is a workaround for AppLocker events using WEF.
Operating Systems Supported for Connector Installation	Linux	Windows, Linux
Event Log Types	Security, System, Application event logs under “Windows Logs” and all event logs under “Applications and Services Logs"	Security, System, Application event logs under “Windows Logs”
Parser Support	Windows OS independent. Windows Event Log connector does not need OS information for correct parsing, so configuring source host OS versions is optional.	Not Windows OS independent

Appendix D: Alternative HTTPS listener creation for older versions of Windows

Create a file with a `.ps1` extension. For example: `ConfigHTTPSList.ps1`. Copy the code below into the file and then run it from a PowerShell prompt. If the script is successful, an HTTPS listener will be created for the WinRM service.

Note: When copying the script, please make sure to remove all the line breaks and paste the script in a single line. This will preserve the reliability and functionality of the script.

As an example, if we remove the line breaks of the next block of code:

```
function
 {
 par1,
 par2,
 par3
 };
 com1;
 com2;
```

The result should be the following:

```
function{par1,par2,par3};com1;com2;
```

If you encounter any issues with the script, please contact support for further assistance.

```
[CmdletBinding()]
Param (
[string]$SubjectName = $env:COMPUTERNAME,
[int]$CerValDay = 1095,
[switch]$SkipNetworkProfileCheck,
$CreateSelfSignedCert = $true,
[switch]$ForceNewSSLCert,
[switch]$GlobalHttpFirewallAccess
);
Function New-LegacySelfSignedCert
{
Param (
```

```
[string]$SubjectName,
[int]$ValidDays = 1095
);
$name = New-Object -COM "X509Enrollment.CX500DistinguishedName.1";
$name.Encode("CN=$SubjectName", 0);
$key = New-Object -COM "X509Enrollment.CX509PrivateKey.1";
$key.ProviderName = "Microsoft RSA SChannel Cryptographic Provider";
$key.KeySpec = 1;
$key.Length = 4096;
$key.SecurityDescriptor =
"D:PAI(A;;0xd01f01ff;;;SY)(A;;0xd01f01ff;;;BA)(A;;0x80120089;;;NS)";
$key.MachineContext = 1;
$key.Create();
$serverauthoid = New-Object -COM "X509Enrollment.CObjectId.1";
$serverauthoid.InitializeFromValue("1.3.6.1.5.5.7.3.1");
$ekuoids = New-Object -COM "X509Enrollment.CObjectIds.1";
$ekuoids.Add($serverauthoid);
$ekuext =
New-Object -COM "X509Enrollment.CX509ExtensionEnhancedKeyUsage.1";
$ekuext.InitializeEncode($ekuoids);
$cert =
New-Object -COM "X509Enrollment.CX509CertificateRequestCertificate.1";
$cert.InitializeFromPrivateKey(2, $key, "");
$cert.Subject = $name;
$cert.Issuer =
$cert.Subject;
$cert.NotBefore = (Get-Date).AddDays(-1);
$cert.NotAfter =
$cert.NotBefore.AddDays($ValidDays);
$cert.X509Extensions.Add($ekuext);
$cert.Encode();
```

```
$enrollment = New-Object -COM "X509Enrollment.CX509Enrollment.1";
$enrollment.InitializeFromRequest($cert);
$certdata = $enrollment.CreateRequest(0);
$enrollment.InstallResponse(2, $certdata, 0, "");
$parsed_cert =
New-Object System.Security.Cryptography.X509Certificates.X509Certificate2;
$parsed_cert.Import([System.Text.Encoding]::UTF8.GetBytes($certdata));
return $parsed_cert.Thumbprint;
}
$listeners = Get-ChildItem WSMan:\localhost\Listener;
If (!($listeners | Where {$_.Keys -like "TRANSPORT=HTTPS"}))
{
$thumbprint =
New-LegacySelfSignedCert -SubjectName $SubjectName -ValidDays $CerValDay;
$valueset = @{
Hostname = $SubjectName;
CertificateThumbprint = $thumbprint;
};
$selectorset = @{
Transport = "HTTPS";
Address = "*";
};
Write-Verbose "Enabling SSL listener.";
New-WSManInstance -ResourceURI 'winrm/config/Listener'
-SelectorSet $selectorset -ValueSet $valueset;
};
```

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 Configuration Guide (SmartConnectors 7.12.0)

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!