# IBM QRadar WinCollect
# 7.3

*WinCollect User Guide V7.3.1*

**Note**

Before you use this information and the product that it supports, read the information in “Notices” on page 99.

# Contents

# About this WinCollect User Guide

This documentation provides you with information that you need to install and configure WinCollect agents, and retrieve events from Windows-based event sources. WinCollect is supported by IBM® Security QRadar® SIEM and IBM QRadar Log Manager.

## Intended audience

System administrators who are responsible for installing WinCollect must be familiar with network security concepts and device configurations.

## Technical documentation

To find IBM Security QRadar product documentation on the web, including all translated documentation, access the IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SS42VS/welcome).

For information about how to access more technical documentation in the QRadar products library, see QRadar Support – Assistance 101 (https://ibm.biz/qradarsupport).

## Contacting customer support

For information about contacting customer support, see QRadar Support – Assistance 101 (https://ibm.biz/qradarsupport).

## Statement of good security practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

# Chapter 1. WinCollect overview

WinCollect is a Syslog event forwarder that administrators can use to forward events from Windows logs to QRadar. WinCollect can collect events from systems locally or be configured to remotely poll other Windows systems for events.

WinCollect is one of many solutions for Windows event collection. For more information about alternatives to WinCollect, see the IBM Security QRadar DSM Configuration Guide.

## How does WinCollect Work?

WinCollect uses the Windows Event Log API to gather events, and then WinCollect sends the events to QRadar.

**Note:** Managed deployment is not supported in QRadar on Cloud environments. Customers who use IBM QRadar on Cloud must use stand-alone WinCollect agents.

## WinCollect managed deployment

A managed WinCollect deployment has a QRadar appliance that shares information with the WinCollect agent that is installed on the Windows hosts that you want to monitor. The Windows host can either gather information from itself, the local host, and, or remote Windows hosts. Remote hosts don't have the WinCollect software installed. The Windows host with WinCollect software installed polls the remote hosts, and then sends event information to QRadar.

**Note:** Managed deployment is not supported in QRadar on Cloud environments. Customers who use IBM QRadar on Cloud must use stand-alone WinCollect agents.

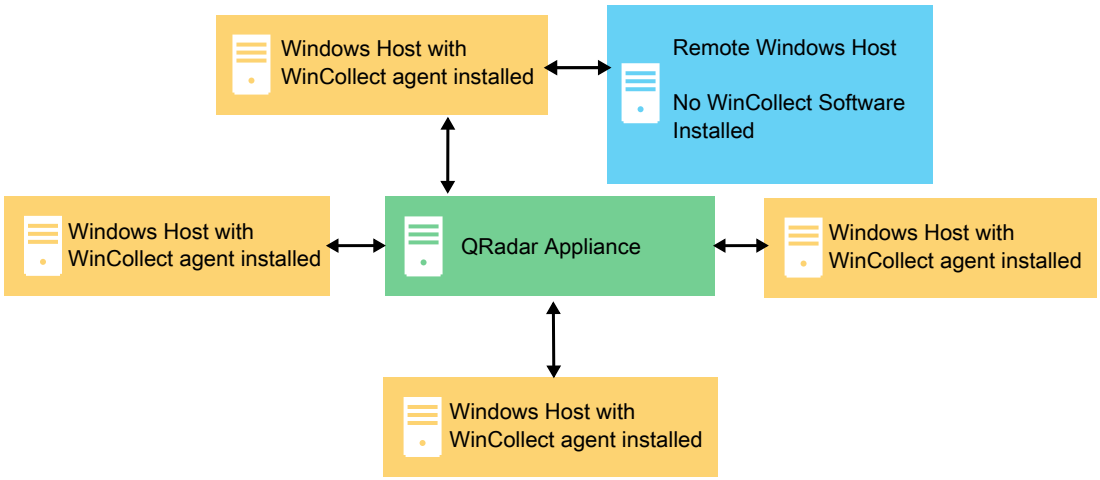



*Figure 1. WinCollect managed deployment example*

**Important:**

1. In a managed deployment, the WinCollect agents that are installed on Windows hosts can be managed by any QRadar Console, Event Collector, or Event Processor.
2. Managed WinCollect deployments are not supported on QRadar on Cloud.

In a managed deployment, WinCollect is designed to work with up to 500 Windows agents per Console and managed host. For example, if you have a deployment with a Console, an Event Processor, and an Event Collector, each can support up to 500 Windows agents, for a total of 1,500. If you want to monitor more than 500 Windows agents per Console or managed host, use the stand-alone WinCollect deployment.

For more information, see “Stand-alone WinCollect Installations” on page 19.

The managed WinCollect deployment has the following capabilities:

- Central management from the QRadar Console or managed host.
- Automatic local log source creation at the time of installation.
- Event storage to ensure that no events are dropped.
- Collects forwarded events from Microsoft Subscriptions.
- Filters events by using XPath queries or exclusion filters.
- Supports virtual machine installations.
- Console can send software updates to remote WinCollect agents without you reinstalling agents in your network.
- Forwards events on a set schedule (Store and Forward)

## WinCollect stand-alone deployment

If you need to collect Windows events from more than 500 agents, use the stand-alone WinCollect deployment. A stand-alone deployment is a Windows host in unmanaged mode with WinCollect software installed. The Windows host can either gather information from itself, the local host, and, or remote Windows hosts. Remote hosts don't have the WinCollect software installed. The Windows host with WinCollect software installed polls the remote hosts, and then sends event information to QRadar. To save time when you configure more than 500 Windows agents, you can use a solution such as IBM Endpoint Manager. Automation can help you manage stand-alone instances.

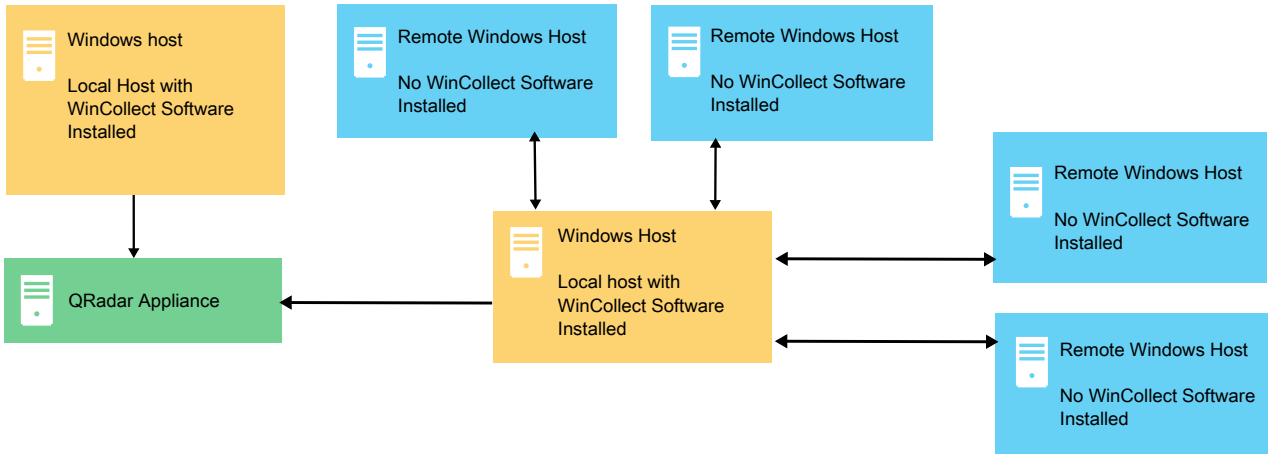



*Figure 2. WinCollect stand-alone deployment example*

You can also deploy stand-alone WinCollect to consolidate event data on one Windows host, where WinCollect collects events to send to QRadar.

Stand-alone WinCollect mode has the following capabilities:

- You can configure each WinCollect agent by using the WinCollect Configuration Console.
- You can update WinCollect software with the software update installer.
- Event storage to ensure that no events are dropped.
- Collects forwarded events from Microsoft Subscriptions.
- Filters events by using XPath queries or exclusion filters.
- Supports virtual machine installations.
- Send events to QRadar using TLS Syslog.
- Automatically create a local log source at the time of agent installation.

## Capabilities of managed and stand-alone WinCollect deployments

Review the following table to understand which capabilities are available when using managed or stand-alone WinCollect agents.

*Table 1. Capabilities of managed WinCollect vs. stand-alone WinCollect*

| Capability | Managed WinCollect | Stand-alone WinCollect |
|---|---|---|
| Central management from the QRadar Console or managed host. | Yes | No |
| Automatic local log source creation at the time of installation. | Yes | Yes |
| Event storage to ensure that no events are dropped. | Yes | Yes |
| Collects forwarded events from Microsoft Subscriptions. | Yes | Yes |
| Filters events by using XPath queries or exclusion filters. | Yes | Yes |
| Supports virtual machine installations. | Yes | Yes |
| QRadar Console can send software updates to WinCollect agents. | Yes | No |
| Forwards events on a set schedule (Store and Forward). | Yes | No |
| You can configure each WinCollect agent by using the WinCollect Configuration Console. | No | Yes |
| You can update WinCollect software with the software update installer. | No | Yes |
| Available with QRadar on Cloud | No | Yes |
| Available with on-prem QRadar | Yes | Yes |

## Setting up a managed WinCollect deployment

For a managed deployment, follow these steps:

1. Understand the prerequisites for managed WinCollect, which ports to use, what hardware is required, how to upgrade. For more information, see Chapter 2, “Installation prerequisites for WinCollect,” on page 7.
2. Install the WinCollect application on the QRadar console. For more information, see “Installing and upgrading the WinCollect application on QRadar appliances” on page 15.
3. Create an authentication token so that the managed WinCollect agents can exchange data with QRadar appliances. For more information, see “Creating an authentication token for WinCollect agents” on page 17.
4. Configure a forwarding destination host for the log source data. For more information, see “Adding a destination” on page 37.
5. Install managed WinCollect agents on the Windows hosts. For more information, see one of the following options:
   - “Installing the WinCollect agent on a Windows host” on page 23
   - “Installing a WinCollect agent from the command prompt” on page 27, or

- "Manually adding a WinCollect agent " on page 35

6. If you want to configure forwarded events or event subscriptions, see "Windows event subscriptions for WinCollect agents" on page 53.
7. If you want to use the legacy Log Source UI to bulk add log sources that will be remotely polled by a single WinCollect agent, see "Bulk log sources for remote event collection" on page 87.
8. Tune your WinCollect log sources. For more information, see the Event Rate Tuning Profile parameter in "Windows log source parameters" on page 58.
9. If you want a managed WinCollect agent to send events to multiple QRadar destinations in case one fails, see "Adding multiple destinations to WinCollect agents" on page 18.

### Setting up a stand-alone WinCollect deployment

For a stand-alone deployment, follow these steps:

1. Understand the prerequisites for stand-alone WinCollect, which ports to use, what hardware is required, how to upgrade. For more information, see Chapter 2, "Installation prerequisites for WinCollect," on page 7.
2. Install stand-alone WinCollect agents on the Windows hosts. For more information, see "Installing the WinCollect agent on a Windows host" on page 23.
3. If you want to add new log sources to your agent or modify existing log sources, install the WinCollect stand-alone configuration console. For more information, see "Installing the configuration console" on page 21 or "Silently installing, upgrading, and uninstalling WinCollect software" on page 22.
4. Configure the destination where the Windows hosts send Windows events. For more information, see "Adding a destination to the WinCollect Configuration Console" on page 41.
5. If you want to use the stand-alone WinCollect agent to collect events from other devices using remote polling, create a credential in the WinCollect stand-alone configuration console, so that WinCollect can log in to the remote devices. For more information, see "Creating a WinCollect credential" on page 41.
6. If you want to add additional log sources to the stand-alone WinCollect agent, do so using the WinCollect stand-alone configuration console. For more information, see "Adding a device to the WinCollect Configuration Console" on page 42.

## What's new in WinCollect

Learn about the new features in each WinCollect release.

### What's new in V7.3.1

**Important:** Managed WinCollect versions 7.3.1 p2 and later are compatible only with QRadar 7.5.0 UP4 and later.

Managed WinCollect versions 7.3.1 p1 (build 22) and earlier are compatible only with QRadar 7.5.0 update package 3 and earlier. Managed WinCollect 7.x users who want to update to QRadar 7.5.0 UP4 must perform an sfs installation of QRadar 7.3.1 p2. For more information, see https://www.ibm.com/support/pages/node/6953887.

**Note:** WinCollect 7.3.1 can only be installed on QRadar 7.3.3 or later.

WinCollect 7.3.1 includes the following capabilities:

- Added a WinCollect configuration server logging protocol that allows more detailed debugging messages.
- You can now reregister an agent with the same name in a managed deployment.
- Improvements made with restoring WinCollect agents in a restored QRadar deployment.

### What's new in V7.3.0

**Note:** WinCollect 7.3.0 can only be installed on QRadar 7.3.3 or later.

WinCollect 7.3.0 includes the following capabilities:

- You can set the Status Server setting to **Disabled** to send only a heartbeat without status messages, or set the value to **None** if you don't want to send a heartbeat or status messages.
- You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

  **Note:** This feature is available for stand-alone deployments. This will be available for Managed agents in a future release of QRadar.

### What's new in 7.2.9

WinCollect 7.2.9 includes the following capabilities:

- Event Forwarding Filtering
- Event Forwarding Sending to one log source support
- Digitally signed installers
- Millisecond Time format for Event Log collection
- DHCP support for Spanish and Polish
- CP Support for Status Messages
- File Forwarder multi-line log support
- Removed MMC requirement from patch installer install

## MSEVEN6 protocol

MSEVEN6 is a Microsoft event protocol that collects more information from an event log, such as the task, keyword, and opcode. It also provides a better message formatting than other event protocols do.

The MSEVEN protocol uses port 445. The NETBIOS ports (137 - 139) can be used for hostname resolution. When the WinCollect agent polls a remote event log by using MSEVEN6, the initial communication with the remote computer occurs on port 135 (dynamic port mapper), which assigns the connection to a dynamic port. The default port range for dynamic ports is between port 49152 and port 65535, but might be different depending on the server type. For example, the default port range for Microsoft Exchange servers is 6005 – 58321.

XPath queries always use the MSEVEN6 event protocol.

In managed mode, you can change the protocol by editing the **Event Log Poll Protocol** field and selecting the desired protocol. For upgrades, depending on which version of WinCollect you are upgrading from, the log source continues to use MSEVEN. Use the Log Source Management app to configure multiple log sources to the desired protocol.

In a stand-alone WinCollect deployment, you can set a global Default Event Log Poll Protocol. The default value is **MSEVEN6**. To configure a single Microsoft Windows Event Log device to use the global Default Event Log Poll Protocol, select **Default** from the **Basic Configurations** page of the device. Otherwise, select **MSEVEN6** or **MSEVEN** to override the global Default Event Log Poll Protocol.

In a stand-alone WinCollect deployment, you can include milliseconds in the time stamp for Event Logs. This option is only compatible in a stand-alone WinCollect deployment that uses the MSEVEN6 protocol. It is not supported by the MSEVEN protocol.

# Chapter 2. Installation prerequisites for WinCollect

Before you can install WinCollect agents, you must verify that your deployment meets the installation requirements.

## Supported versions

Administrators should be aware that supported software versions for IBM WinCollect is the Latest version (n) and latest minus one (n-1). This means that the two newest versions of WinCollect are the versions for which QRadar Support will provide full support with any support tickets (cases) that are opened. Customers using older versions of WinCollect will receive minimal, best effort, support. To prevent issues, it is important that administrators keep WinCollect deployments updated when new versions are posted to IBM Fix Central.

**Note:** WinCollectdoes not support agents installed on Windows servers that use Network Address Translation (NAT). If you place an Event Collector in the same NAT environment as the managed agents, the agents can use the Event Collector as a configuration server, status server, and to send events. However, the Event Collector must be configured to use NAT.

## Distribution options for WinCollect agents

WinCollect agents can be distributed in a remote collection configuration or installed on the local host.

**Local collection**

The WinCollect agent collects events only for the host on which it is installed. You can use this collection method on a Windows host that is busy or has limited resources, for example, domain controllers.

**Important:** QRadar Support recommends local collection on Domain Controllers and other high EPS servers, as it is more stable than remote collection. If you are remote polling logs on potentially high EPS servers, QRadar Support might require you to install an agent locally on the server.

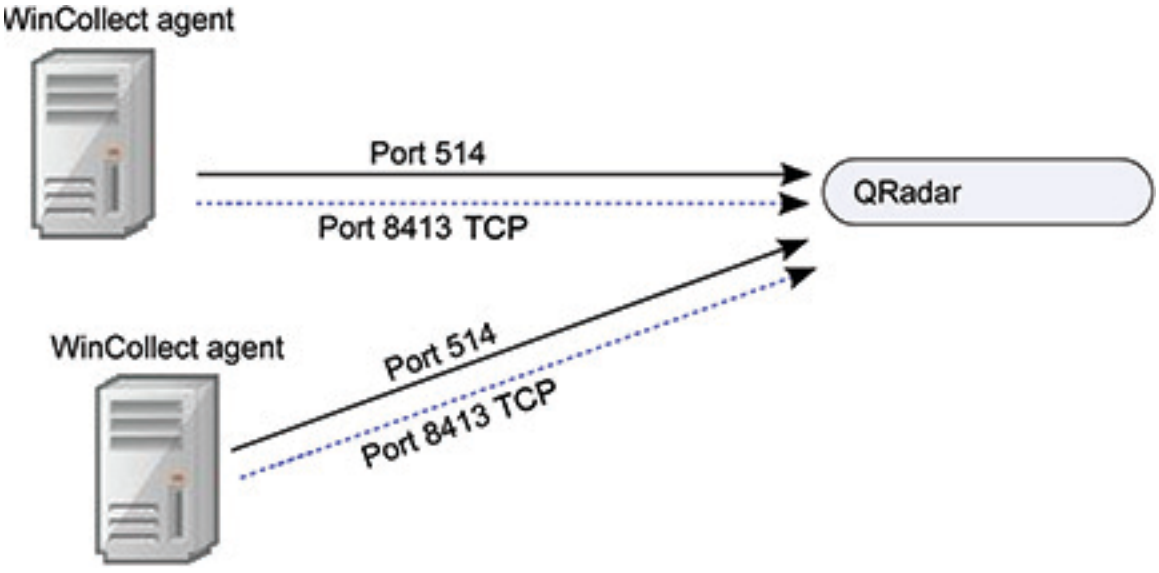



*Figure 3. Local collection for WinCollect agents*

**Remote Collection**

The WinCollect agent is installed on a single host and collects events from multiple Windows systems. Use remote collection to easily scale the number of Windows log sources that you can monitor.

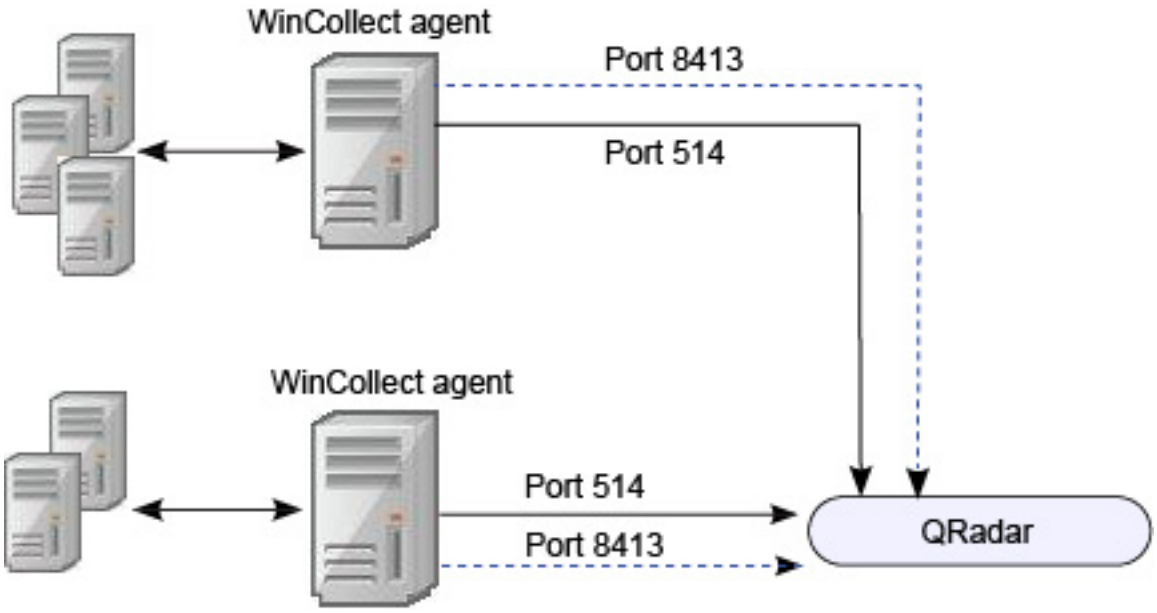



*Figure 4. Remote collection for WinCollect agents*

# Communication between WinCollect agents and QRadar

Open ports are required for data communication between WinCollect agents and the QRadar host, and between WinCollect agents and the hosts that they remotely poll.

## WinCollect agent communication to QRadar Console and Event Collectors

All WinCollect agents communicate with the QRadar Console and Event Collectors to forward events to QRadar and request updated information. Managed WinCollect agents also request and receive updated code and configuration changes. You must ensure firewalls that are between the QRadar Event Collectors and your WinCollect agents allow traffic on the following ports:

**Port 8413**

This port is used for managing the WinCollect agents to request and receive code and configuration updates. Traffic is always initiated from the WinCollect agent, and is sent over TCP. Communication is encrypted by using the QRadar Console's public key and the `ConfigurationServer.PEM` file on the agent.

Create a bidirectional rule to allow communication from the WinCollect agent to QRadar on port 8413. If the rule is not bidirectional, traffic is blocked. QRadar does not send updates to the WinCollect agent on port 8413.

**Port 514**

This port is used by the WinCollect agent to forward syslog events to QRadar. You can configure WinCollect log sources to provide events by using TCP or UDP. You can decide which transmission protocol to use for each WinCollect log source. Port 514 traffic is always initiated from the WinCollect agent.

## WinCollect agents remotely polling Windows event sources

WinCollect agents that remotely poll other Windows operating systems require extra ports to be open. These ports need to be open on the WinCollect agent computer and the computer(s) that are remotely polled, but not on your QRadar appliances. The following table describes the ports that are used.

| *Table 2. Port usage for WinCollect remote polling* | | |
|---|---|---|
| **Port** | **Protocol** | **Usage** |
| 135 | TCP | Microsoft Endpoint Mapper |
| 137 | UDP | NetBIOS name service |
| 138 | UDP | NetBIOS datagram service |
| 139 | TCP | NetBIOS session service |
| 445 | TCP | Microsoft Directory Services for file transfers that use Windows share |
| 49152 – 65535<br>**Note:** Exchange servers are configured for a port range of 6005 – 58321 by default. | TCP | Default dynamic port range for TCP/IP |

The MSEVEN protocol uses port 445. The NETBIOS ports (137 - 139) can be used for host name resolution. When the WinCollect agent polls a remote event log by using MSEVEN6, the initial communication with the remote machine occurs on port 135 (dynamic port mapper), which assigns the connection to a dynamic port. The default port range for dynamic ports is between port 49152 and port 65535, but might be different dependent on the server type. For example, Exchange servers are configured for a port range of 6005 – 58321 by default.

To allow traffic on these dynamic ports, enable and allow the two following inbound rules on the Windows server that is being polled:

- Remote Event Log Management (RPC)
- Remote Event Log Management (RPC-EPMAP)

**Important:** To limit the number of events that are sent to QRadar, administrators can use exclusion filters for an event based on the EventID or Process. For more information about WinCollect filtering, see WinCollect Event Filtering (http://www.ibm.com/support/docview.wss?uid=swg21672656).

**Related concepts**

"MSEVEN6 protocol" on page 5

MSEVEN6 is a Microsoft event protocol that collects more information from an event log, such as the task, keyword, and opcode. It also provides a better message formatting than other event protocols do.

# Enabling remote log management on Windows

You can enable remote log management only when your log source is configured to remotely poll other Windows operating systems. You can enable remote log management on Windows 2012 R2 for XPath queries.

## About this task

**Note:** WinCollect does not support reverting Citrix Virtual Machines that are polled remotely.

## Procedure

1. On your desktop, select **Start** > **Control Panel**.
2. Click the **System and Security** icon.
3. Click **Allow a program through Windows Firewall**.
4. If prompted, click **Continue**.
5. Click **Change Settings**.
6. From the **Allowed programs and features** pane, select **Remote Event Log Management**.

Depending on your network, you might need to correct or select more network types.

7. Click **OK**.

# Hardware and software requirements for the WinCollect host

Ensure that the Windows-based computer that hosts the WinCollect agent meets the minimum hardware and software requirements.

## Hardware/virtual machine requirements

The following table describes the minimum hardware requirements for local collection:

*Table 3. Hardware/VM requirements for local collection by using WinCollect*

| Requirement | Description |
|---|---|
| Memory | The WinCollect agent has a very low memory footprint. The following numbers were generated on virtual machines (VMs) with two Logical cores and 2-4GB of memory.<br>1 Event per second (EPS) or less: 9 MB<br>100 EPS or less: 10.5 MB<br>2,500 EPS or less: 15 MB<br>5,000 EPS or less: 20 MB |
| Processor | Intel Core i3 or equivalent<br>Systems were tested on VMs with two Cores and 2 - 4 GB of memory. |
| Available processor resources | 0-35%, depending on CPU, EPS, and number of endpoints polled. See the following table for examples.<br>Very high EPS rates have a direct effect on the Average CPU used by the WinCollect Agent. |
| Disk space | 100 MB for software, plus up to 100 MB for files.<br>Up to 6 GB might be required if you store events to disk. |

**Note:** WinCollect CPU and memory loads depend on several factors, including the number of events per second that are being processed.

The following table shows resources that are used by WinCollect in testing environments with various hardware configurations and EPS counts.

*Table 4. Comparison of tested WinCollect environments (local polling)*

| Profile | Type | OS | RAM | Cores | Avg EPS | RAM used | Avg CPU |
|---|---|---|---|---|---|---|---|
| Maximum EPS | VM | Windows 2019 Server | 4 GB | 2 | 5,000 | 20 MB | 32% |
| High EPS | VM | Windows 2019 Server | 4 GB | 2 | 2,500 | 15 MB | 18% |

| *Table 4. Comparison of tested WinCollect environments (local polling) (continued)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Profile** | **Type** | **OS** | **RAM** | **Cores** | **Avg EPS** | **RAM used** | **Avg CPU** |
| Medium EPS | VM | Windows 2019 Server | 4 GB | 2 | 100 | 10.5 MB | 1.2% |
| Low EPS | VM | Windows 2019 Server | 4 GB | 2 | <1 | 9 MB | <1% |

Similar results were found testing with Windows 2016 Server.

A lower provisioned Windows 10 VM yielded similar results.

| *Table 5.* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Profile** | **Type** | **OS** | **RAM** | **Cores** | **Avg EPS** | **RAM used** | **Avg CPU** |
| High EPS | VM | Windows 10 | 2 GB | 2 | 2500 | 11 MB | 22% |
| Medium EPS | VM | Windows 10 | 2 GB | 2 | 100 | 5.5 MB | 1.5% |
| Low EPS | VM | Windows 10 | 2 GB | 2 | <1 | 5.5 MB | <1 |

The following table describes the minimum hardware requirements for remote collection:

| *Table 6. Hardware/VM requirements for remote collection by using WinCollect* | |
|---|---|
| **Requirement** | **Description** |
| Memory | 5 endpoints or less: 80 MB<br>250 endpoints or less: 293 MB<br>500 endpoints or less: 609 MB |
| Processor | Intel Core i3 or equivalent |
| Available processor resources | Approximately 20%, depending on CPU, EPS, and number of endpoints polled. |
| Disk space | 100 MB for software, plus up to 100 MB for files.<br>Up to 6 GB might be required if you store events to disk. |

**Note:** WinCollect CPU and memory loads depend on several factors, including the number of events per second that are being processed and the number of remote endpoints that are being polled.

| *Table 7. Comparison of tested WinCollect environments (remote polling)* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Profile** | **Type** | **OS** | **RAM** | **Cores** | **Endpoints polled** | **Avg EPS** | **RAM used** | **Avg CPU** |
| High EPS Low Device Count | VM | Windows 2012 Server | 12 GB | 8 | 6 | 3,000 | 78 MB | 6.5% |

| Table 7. Comparison of tested WinCollect environments (remote polling) (continued) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Profile** | **Type** | **OS** | **RAM** | **Cores** | **Endpoints polled** | **Avg EPS** | **RAM used** | **Avg CPU** |
| Medium EPS and Device count | VM | Windows 2016 Server | 12 GB | 4 | 250 | 2,500 | 290 MB | 14% |
| High EPS High Device count | VM | Windows 2016 Server | 16 GB | 8 | 500 | 5,000 | 605 MB | 10.75% |

### Software requirements

The following table describes the software requirements:

| Table 8. Software requirements | |
|---|---|
| **Requirement** | **Description** |
| Operating system | Windows Server 2022 (including Core)<br>Windows Server 2019 (including Core)<br>Windows Server 2016 (including Core)<br>Windows Server 2012 (including Core)<br>Windows 10 |
| Distribution | One WinCollect agent for each Windows host. |
| Required user role permissions for installation | Administrator, or local administrator<br>Administrative permissions are not required for remote collection. |

**Important:** WinCollect is not supported on versions of Windows that are designated end-of-life by Microsoft. After software is beyond the Extended Support End Date, the product might still function as expected. However, IBM does not make code or vulnerability fixes to resolve WinCollect issues for older operating systems. For example, Microsoft Windows Server 2003 R2 and Microsoft Windows XP are operating systems that are beyond the "Extended Support End Date." Any questions about this announcement can be discussed in the IBM QRadar Collecting Windows Events (`WMI/ALE/WinCollect`) forum. For more information, see https://support.microsoft.com/en-us/lifecycle/search (https://support.microsoft.com/en-us/lifecycle/search).

## Prerequisites for upgrading WinCollect agents in a managed deployment

Before you upgrade WinCollect agents, ensure that your software meets the version requirements.

### WinCollect and QRadar software versions

The version of the installed WinCollect depends on the version of QRadar that you are running.

| *Table 9. Software version matrix* | | |
|---|---|---|
| **QRadar Version** | **Minimum WinCollect Version** | **RPM Minimum Version** |
| QRadar V7.3.x | WinCollect 7.2.5 | AGENT-WINCOLLECT-7.3-20161123160813.noarch |
| QRadar V7.4.x | WinCollect 7.2.5 | AGENT-WINCOLLECT-7.3-20161123160813.noarch |
| QRadar V7.5.x | WinCollect 7.2.5 | AGENT-WINCOLLECT-7.3-20161123160813.noarch |

## Checking the installed version of the WinCollect agent

You can check the version of the installed WinCollect agent by performing the following steps:

1. In QRadar, select **Help** > **About**
2. Select the **Additional Release Information** link.
3. If you want to verify the WinCollect agent release, use ssh to log in to the QRadar Console as the root user, and run the following command:

```
yum list all | grep -i AGENT-WINCOLLECT
```

# Chapter 3. WinCollect installations

You install WinCollect agents in an environment that is managed by IBM QRadar, or as a stand-alone agent, or a combination of both.

## Managed WinCollect installations

To use managed WinCollect, you must download and install a WinCollect Agent SF Bundle on your QRadar console, create an authentication token, and then install a managed WinCollect agent on each Windows host that you want to collect events from. You can also install the managed WinCollect agent on a Windows host that you want to use to remotely collect events from other Windows hosts.

**Note:** Stand-alone WinCollect, the type of deployment used by IBM QRadar on Cloud, does not require you to download and install a WinCollect Agent SFS Bundle on your QRadar console, nor does it require you to create a WinCollect authentication token.

**Important:** WinCollect does not support cloning virtual machines (VMs) that have agents installed that are registered in QRadar.

### Installing and upgrading the WinCollect application on QRadar appliances

To manage a deployment of WinCollect agents from the QRadar user interface, you must first upgrade your QRadar Console to a supported version of WinCollect by using the WinCollect Agent SFS Bundle. This bundle includes the required protocols to enable communication between QRadar and the managed WinCollect agents on the Windows hosts. Both the QRadar Console and managed WinCollect agents can be upgraded to newer versions of WinCollect by installing the newer version of SFS Bundle on the QRadar console.

#### About this task

**Important:**

- For information about upgrading WinCollect versions v7.0 through v7.2.2, see www.ibm.com/support (http://www-01.ibm.com/support/docview.wss?uid=swg21698127).
- If WinCollect v7.2.6 or newer is installed on the QRadar Console, and then you upgrade QRadar from v7.2.8 to v7.3.0 or newer, the version of WinCollect on QRadar reverts to v7.2.5. The managed WinCollect agents that are running on your Windows hosts remain at their current version and continue to send events to QRadar using their existing configuration information. However, they no longer receive code or configuration updates. You must reinstall a version of the WinCollect Agent SFS Bundle that is the same as or newer than your current agents' version on your QRadar Console after the QRadar upgrade.

After you upgrade a QRadar Console, the managed WinCollect agents that are enabled to receive automatic updates automatically upgrade to the new version of WinCollect at the next configuration polling interval. If new WinCollect agent files are available for download, the agent downloads, installs updates, and restarts required services. No events are lost when you update your WinCollect agent because events are buffered to disk. Event collection forwarding continues when the WinCollect service on the Windows host restarts.

**Important:** If you reinstall QRadar on your Console, you must delete this file on any existing WinCollect agent installations before WinCollect can function properly: `Program Files/IBM/WinCollect/config/ConfigurationServer.PEM`

#### Procedure

1. Download the WinCollect Agent SFS Bundle installation file from the IBM website: (http://www.ibm.com/support).

**Note:** The installation process restarts services on the Console, which creates a gap in event collection until services restart. Schedule the WinCollect upgrade during a maintenance window to avoid disrupting users.

2. Use SSH to log in to the QRadar Console as the root user.
3. For initial installations, create the `/storetmp` and `/media/updates` directories if they do not exist. Type the following commands:

```
mkdir /media/updates
mkdir /storetmp
```

4. Using a program such as `WinSCP`, copy the downloaded SFS file to `/storetmp` on your QRadar Console.
5. To change to the `/storetmp` directory, type the following command: `cd /storetmp`
6. To mount the SFS file, type the following command: `mount -t squashfs -o loop` *`<Installer_file_name.sfs>`* `/media/updates`

   **Example:** `mount -t squashfs -o loop 730_QRadar_wincollectupdate-7.3.0-24.sfs /media/updates`
7. To run the WinCollect installer, type the following command and then follow the prompts: `/media/updates/installer`

   **Note:** To proceed with the WinCollect Agent update you must restart services on QRadar to apply protocol updates. The following message is displayed:

```
WARNING: Services need to be shutdown in order to apply patches.
This will cause an interruption to data collection and correlation.
Do you wish to continue (Y/N)?
```

8. Type Y to continue with the update.

   During the update, the SFS installs new protocol updates. If your Secure Shell (SSH) session is disconnected while the upgrade is in progress, the upgrade continues. When you reopen your SSH session and run the installer again, the patch installation resumes. After the installation is complete, services are restarted, and the user interface is available.

   **Note:** During installation, the following message is displayed:

```
Patch 144249
This patch includes a new version of the WinCollect Configuration Server.
For this new version to run properly, the event collection service needs to be restarted.
If you choose to not restart the service, agents cannot get new configurations and code
updates until you restart it.

Choices:
1. Restart event collection service at the end of the patch installation, on the Console
and on all managed hosts patched from the Console.
2. Do not restart event collection service yet. You will need to restart it in the user
interface (Advanced > Restart Event Collection Services).
3. Abort patch.
```

   After you choose an option, the patch installation continues. When it is complete, press the Enter key to exit the patch screen.
9. If you selected the second option in step 8, you must perform the following steps:
   - In the QRadar admin settings, click **Advanced** > **Deploy Full Configuration**.
   - In the QRadar admin settings, click **Advanced** > **Restart Event Collection Services**.
10. To unmount the SFS file from the Console, type the following command: `umount /media/updates`
11. Verify that WinCollect agents are configured to accept remote updates:
    a) Log in to QRadar.
    b) On the navigation menu, click **Data Sources**.
    c) Click the WinCollect icon.

d) Review the **Automatic Updates Enabled** column and select WinCollect agents that have a False value.

e) Click **Enable/Disable Automatic Updates**.

### Results

Managed WinCollect agents with automatic updates enabled are updated and restarted. The amount of time it takes a managed agent to update depends on the configuration polling interval for the agent and the speed of the network connections between the Console and the agent.

**Related tasks**

Installing the WinCollect agent on a Windows host

Installing a WinCollect agent from the command prompt

For unattended installations, you can install the WinCollect agent from the command prompt. Use the silent installation option to deploy WinCollect agents simultaneously to multiple remote systems.

## Creating an authentication token for WinCollect agents

Third-party or external applications that interact with IBM Security QRadar require an authentication token. Before you install managed WinCollect agents in your network, you must create an authentication token.

An authentication token is not required for stand-alone WinCollect agents, such as those used in IBM QRadar on Cloud, but every managed WinCollect agent must use an authentication token.

The authentication token allows managed WinCollect agents to exchange data with QRadar appliances. Create one authentication token to use for all of your managed WinCollect agents that communicate events with QRadar. If the authentication token expires, the WinCollect agent cannot receive log source configuration changes or code updates.

### About this task

**Note:** This capability is not available in IBM QRadar on Cloud.

### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **System Configuration**.
3. Click the **Authorized Services** icon.
4. Click **Add Authorized Service**.
5. In the **Manage Authorized Services** window, configure the parameters.

*Table 10. **Add Authorized Services** parameters*

| Parameter | Description |
|---|---|
| **Service Name** | The name can be up to 255 characters in length, for example, `WinCollect Agent`. |
| **User Role** | Select **WinCollect**.<br>For more information about user roles, see the *IBM Security QRadar SIEM Administration Guide*. |
| **Expiry** | Select **No Expiry**. |

6. Click **Create Service**.
7. Record the token value.

# Adding multiple destinations to WinCollect agents

In a managed WinCollect deployment, add IBM QRadar appliances as destinations for Windows events if a QRadar appliance fails.

## Before you begin

You must create the destinations that you want to add to the WinCollect agent. See "Adding a destination" on page 37.

## About this task

Each destination that you create for a WinCollect agent has its own disk cache for events. If Site A fails and Site B is configured as the Target External Destination, Site B continues to receive events and Site A stores events to disk. If both sites fail, both systems are caching events independently to separate disk queues. As connections return for individual log sources, the agents attempt to balance sending new events and cached events that are queued due to either bursting events, or connection issues.

If your deployment contains many log sources by using multiple destinations, increase the default disk space. Each agent is configured with 6 GB of disk space to cache events. However, if there are 50 log sources or more, each sending to multiple destinations, and a network segment fails, each log source writes two sets of events to the same cache on the Target Internal and the Target External destination. If your deployment contains segments that are unstable or a prone to outages, update the default storage capacity of the agent in the event of a long term outage.

## Procedure

1. In QRadar, click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Click **Agents** and select the agent that you want to edit.
5. Click **Log Sources**.
6. Select the log source that you want to edit, and click **Edit**.
7. Select the **Target External Destinations** check box.
8. Select the destinations that you want to add to the agent from the box below the **Target External Destinations** check box.
9. Click **Save**.

# Migrating WinCollect agents after a QRadar hardware upgrade

After a QRadar hardware upgrade, you need to generate a new authorization token for your WinCollect agents and update their `install_config` files.

## About this task

## Procedure

1. Generate an authentication token. For more information, see "Creating an authentication token for WinCollect agents" on page 17.

   **Note:** This capability is not available in IBM QRadar on Cloud.
2. Stop the WinCollect agent service.
3. Delete the certificate file `C:\Program Files\IBM\WinCollect\config\ConfigurationServer.PEM`
4. Update the `\WinCollect\config\install_config.txt` file with the IP address of your new Console.

5. Run the following command, where *<auth_token>* is the authentication token that you generated in step 1:

   ```
   C:\Program Files\IBM\WinCollect\bin\InstallHelper.exe -T <auth_token> -a
   "C:\Program Files\IBM\WinCollect\config\install_config_autocreate.txt"
   ```

   ```
   C:\Program Files\IBM\WinCollect\bin\InstallHelper.exe -T
   xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx -a "C:\Program
   Files\IBM\WinCollect\config\install_config_autocreate.txt"
   ```

6. Restart the WinCollect agent.

## Migrating from Adaptive Log Exporter to WinCollect

To migrate from Adaptive Log Exporter (ALE) deployments to WinCollect, install the WinCollect agent, create a log source, and decommission ALE on the Windows host. The ALE product is end of life (EOL), and is no longer supported.

### Procedure

1. Install the WinCollect SFS on the IBM QRadar SIEM Console.
2. Click the **Admin** tab.
3. From the **Data Sources**, click **Wincollect**.
4. On the **WinCollect** page, create a WinCollect destination by clicking **Destinations** > **Add**.
5. Install the WinCollect agent on the Windows host. For more information, see "Installing the WinCollect agent on a Windows host" on page 23.

   **Note:** You can create a log source from the WinCollect installation wizard.
6. Wait for the WinCollect agents to auto discover.
7. Optional. Create a WinCollect log source in QRadar to replace the existing log source that is used by the Adaptive Log Exporter. For more information, see "Adding a log source to a WinCollect agent" on page 87.

   **Note:** You can skip step 7 if **Create Log Source** was selected during the installation of WinCollect. Log sources that use the WinCollect protocol can be created individually or added in bulk for WinCollect agents that remotely poll for events.
8. In the **Log Activity** tab, verify that events are received.
9. Decommission the Adaptive Log Exporter:
   a) Close all active applications on the Windows host.
   b) Open the Windows command prompt.
   c) Go to the installation directory for the Adaptive Log Exporter.

      **Note:** ALE standard installation directory is the `Program Files` or `Program Files (x86)` directory.
   d) To uninstall the Adaptive Log Exporter, type the following command:

      ```
      unins000.exe /SILENT /VERYSILENT
      ```

# Stand-alone WinCollect Installations

A stand-alone deployment is a Windows host in unmanaged mode with WinCollect software installed. The Windows host can either gather information from itself, the local host, and, or remote Windows hosts. Remote hosts don't have the WinCollect software installed. The Windows host with WinCollect software installed polls the remote hosts, and then sends event information to IBM QRadar.

## WinCollect Configuration Console overview

In stand-alone deployments, use the WinCollect Configuration Console to manage your WinCollect deployment. Use the WinCollect Configuration Console to add devices that you want WinCollect to collect agents from, and add the IBM QRadar destination where you want to send events.

**Prerequisites:** Before you can install the WinCollect Collect Configuration Console, you must do the following:

- Install the WinCollect agent in stand-alone mode. For more information, see "Installing the WinCollect agent on a Windows host" on page 23.
- Install .net framework version 3.5
- Install Microsoft Management Console (MMC) 3.0 and later.

The following table describes the WinCollect Configuration Console.

*Table 11. WinCollect Configuration Console window*

| Sections | Description |
|---|---|
| **Global Configuration** | The Global Configuration parameter allows you to view, add and update information about the system where WinCollect data is stored. |
| | **Disk Manager** - the path to the WinCollect Data, which is used to buffer events to disk when the event rate exceeds the event throttle.<br>**Capacity** is the maximum capacity allowed for the contents of the Data Folder. WinCollect does not write to this folder after the maximum capacity is reached. |
| | **Installation Information** - displays information about the WinCollect agent installation.<br>**Application Identifier** - the header of the payload messages sent to the status server.<br>**Status Server** - where the WinCollect Agent status events, such as heart beat messages and any warnings or errors generated by the WinCollect Agent, are sent. |
| | **Security Manager** - centralized credentials, used to collect events from remote devices. |
| **Destinations** | The **Destinations** parameter defines where WinCollect device data is sent. |
| | **Syslog TCP** or **Syslog UDP** destinations include the following parameters:<br>**Name**<br>**Hostname**<br>**Port**<br>**Throttle (events per second)**<br>You can expand a destination to view all devices that are assigned to the destination. |

| Table 11. WinCollect Configuration Console window (continued) | |
|---|---|
| **Sections** | **Description** |
| **Devices** | The **Device** parameter contains available device types. Under each device types, you can view or update multiple device parameters. |

# Installing the configuration console

Download and install the WinCollect configuration console to manage your stand-alone deployment. You can choose an option to install just the WinCollect patch, if you are deploying WinCollect on a large number of Windows hosts that do not require the configuration console.

## Before you begin

- The existing WinCollect agent must be in stand-alone mode before you can install the configuration console. For more information about WinCollect agent installations, see "Installing a WinCollect agent from the command prompt" on page 27.
- .NET framework 3.5 features are required. For information about how to verify .NET installations, see www.ibm.com/support (https://www.ibm.com/support/docview.wss?uid=swg21701063).
- Microsoft Management Console (MMC) 3.0 and later is required.
- The WinCollect Stand-alone patch installer supports the following Windows software versions:
  - Windows Server 2019
  - Windows Server 2016
  - Windows Server 2012 (most recent)
  - Windows 10 (most recent)
  - Windows 8 (most recent)
  - Windows Vista (most recent)

**Important:** WinCollect is not supported on versions of Windows that are designated end-of-life by Microsoft. After software is beyond the Extended Support End Date, the product might still function as expected. However, IBM does not make code or vulnerability fixes to resolve WinCollect issues for older operating systems. For example, Microsoft Windows Server 2003 R2 and Microsoft Windows XP are operating systems that are beyond the "Extended Support End Date." Any questions about this announcement can be discussed in the IBM QRadar Collecting Windows Events (`WMI/ALE/WinCollect`) forum. For more information, see https://support.microsoft.com/en-us/lifecycle/search (https://support.microsoft.com/en-us/lifecycle/search).

## Procedure

1. Download the patch software from IBM Support (www.ibm.com/support/fixcentral). onto the Windows host where you want to install the configuration console.
2. Open the executable file on your system.
3. Follow the steps in the installation wizard. You can select an option to install both the WinCollect configuration console, and the WinCollect patch, or just the patch.

## Silently installing, upgrading, and uninstalling WinCollect software

Enter a command to complete all installation and upgrading tasks for the WinCollect stand alone patch, and the WinCollect Configuration Console, rather than using the installation wizard. You can also upgrade WinCollect agents by using the agent installer only.

### Procedure

1. Download the patch software from IBM Support (www.ibm.com/support/fixcentral).
2. Install or upgrade both the WinCollect stand alone patch and the WinCollect Configuration Console by using the following commands:

```
<setup.exe> /s /v" /qn"
```

3. Change the installation directory of the WinCollect Configuration Console by using the following command:

```
<setup.exe> /s /v" /qn ADDLOCAL=ALL INSTALLDIR=<PATH>"
```

4. Install or upgrade only the WinCollect stand-alone patch by using the following command:

```
<setup.exe> /s /v" /qn ADDLOCAL=WinCollect_StandAlone_Patch"
```

5. If you want to uninstall the WinCollect Configuration Console, use the following command:

```
<setup.exe> /s /x /v" /qn"
```

For more information about stand-alone installs, see IBM Support (www.ibm.com/support/docview.wss?uid=swg21698381).

## Setting an XPath parameter during automated installation

In WinCollect V 7.2.8 and later, you can add an XPath parameter to your command line installer for stand-alone WinCollect agent installations.

### Procedure

1. Convert your XPath to base64 encoding using https://www.base64encode.org/ or another encoding tool.

   For example, this XPath, needed to collect Windows PowerShell logs:

```
<QueryList>
<Query Id="0" Path="Windows PowerShell">
<Select Path="Windows PowerShell">*</Select>
</Query>
</QueryList>
```

   results in this base64 conversion:
   PFF1ZXJ5TGlzdD4KPFF1ZXJ5IElkPSIwIiBQYXRoPSJXaW5kb3dzIFBvd2VyU2hlbGwiPgo8U2VsZWN0IFBhdGg9IldpbmRvd3MgUG93ZXJTaGVsbCI+KjwvU2VsZWN0Pgo8L1F1ZXJ5Pgo8L1F1ZXJ5TGlzdD4=

2. Add the following code to your command line installer:

```
c:\wincollect-7.2.8-91.exe /s /v"/qn STATUSSERVER=<valid IP address>
LOG_SOURCE_AUTO_CREATION_
ENABLED=True
LOG_SOURCE_AUTO_CREATION_PARAMETERS=""Component1.AgentDevice=DeviceWindowsLog&Component1.
Action=create&
Component1.LogSourceName=%COMPUTERNAME%&Component1.LogSourceIdentifier=%COMPUTERNAME%&
Component1.Dest.Name=QRadar&Component1.EventLogPollProtocol=MSEVEN6&Component1.Dest.Hostname=
<valid IP address>&
Component1.Dest.Port=514&Component1.Dest.Protocol=TCP&Component1.Log.Security=true&Component1
.Log.System=true&
Component1.Log.Application=true&Component1.Log.DNS+Server=false&Component1.Log.File+Replicati
on+
```

```
Service=false&
Component1.Log.Directory+Service=false&Component1.RemoteMachinePollInterval=3000&
Component1.MinLogsToProcessPerPass=1250&Component1.MaxLogsToProcessPerPass=2500&
Component1.CustomQuery.Base64=<base64 Xpath>&
Component1.EventRateTuningProfile=High+Event+Rate+Server"""
```

**Note:** Replace the following entries with valid IP addresses:

```
STATUSSERVER=<valid IP address>
Component1.Dest.Hostname=<valid IP address>
```

STATUSSERVER is the location where the WinCollect agent sends status messages (such as WinCollect service starting or any agent error messages). `Component1.Dest.Hostname` is the location where the agent sends event logs (such as QRadar EC or Console).

**Note:** Replace the following entry with the base64 conversion you created in Step 1:

```
Component1.CustomQuery.Base64=<base64 Xpath>
```

3. Add or remove any of the Components or event logs you want to collect.

# Installing the WinCollect agent on a Windows host

Install the WinCollect agent on each Windows host that you want to use for local or remote collection in your network environment.

## Before you begin

Ensure that the following conditions are met:

- You created an authentication token for the managed WinCollect agent.

  **Note:** An authentication token is not required for stand-alone WinCollect deployments such as those used in IBM QRadar on Cloud, but every managed WinCollect agent must use an authentication token.

  For more information, see “Creating an authentication token for WinCollect agents” on page 17.
- Your system meets the hardware and software requirements.

  For more information, see “Hardware and software requirements for the WinCollect host” on page 10.
- The required ports are available for WinCollect agents to communicate with QRadar and remotely polled Windows computers.

  For more information, see “Communication between WinCollect agents and QRadar” on page 8.
- To automatically create a log source for a managed WinCollect agent, you must first create a destination that your agent can use to connect to QRadar and create your log source. For more information, see “Adding a destination” on page 37.

  The managed WinCollect agent sends the Windows event logs to the configured destination. The destination can be the QRadar Console, an Event Processor, or an Event Collector.

## Procedure

1. Download the WinCollect Agent `.exe` file from the IBM Support website (http://www.ibm.com/support).
2. Right-click the WinCollect Agent `.exe` file and select **Run as administrator**.
3. Follow the prompts in the installation wizard and use the following parameters for either managed or stand-alone agent setup.

| *Table 12. WinCollect Managed agent setup type installation wizard parameters* | |
|---|---|
| **Parameter** | **Description** |
| **Host Identifier** | Use a unique identifier for each WinCollect agent that you install. The name that you type in this field is displayed in the WinCollect agent list of the QRadar Console. If you are reinstalling an agent on a Windows host and you want to use the same Host Identifier for the agent, you must first rename the existing agent in QRadar. Host identifiers are unique to each installation of the agent on the same Windows host.<br>By default, the Host Identifier is the hostname of the Windows host. |
| **Authentication Token** | The authentication token that you created in QRadar, for example, `af111ff6-4f30-11eb-11fb-1fc117711111`. |
| **Configuration Server (host and port)** | The IP address or host name of your QRadar Console, Event Collector, or Event processor. For example, `192.0.2.0` or `myhost`. |
| **Create Log Source** | If this check box is selected, you must provide information about the log source and the target destination. |
| **Log Source Name** | The name can be a maximum of 255 characters. |
| **Log Source Identifier** | Identifies the device that the WinCollect agent polls. This field must use the hostname, IP address, or FQDN of the Windows host that the log source gathers events from. |
| **Target Destination** | The WinCollect destination must be configured in QRadar before you continue entering information in the installation wizard. This field must contain the name of a previously created WinCollect Destination as it appears in the **Destinations** window. |
| **Event Logs** | The Windows logs that you want the log source to collect events from and send to QRadar. |
| **Machine poll interval (msec)** | The polling interval that determines the number of milliseconds between queries to the Windows host.<br>The minimum polling interval is 300 milliseconds. The default is 3000 milliseconds or 3 seconds. |

| *Table 12. WinCollect Managed agent setup type installation wizard parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Event Rate Tuning Profile** | Select the tuning profile:<br>• Default (Endpoint): 100/150<br>This setting is suitable for Windows endpoints that are running a non-Server OS.<br>• Typical Server: 500/750<br>This setting is suitable for most Windows Server endpoints.<br>• High Event Rate Server: 1250/1875<br>This setting is suitable for all Windows endpoints and is ideal for Domain Controllers and other potentially high EPS endpoints.<br>For more information, see IBM Support (http://www-01.ibm.com/support/docview.wss?uid=swg21672193). |
| **Default Status Server Address** | An alternative destination to send WinCollect status messages to, such as the heartbeat, if required. Set the value to an IP address to send status messages to any QRadar Console or any Event Processor or Event Collector in your deployment. Set the value to `Disabled` to send only a heartbeat without status messages. Set the value to None if you don't want to send a heartbeat or status messages. |
| **Syslog Status Server (if different from default)** | An alternative destination to send WinCollect status messages to, such as the heartbeat, if required. Set the value to an IP address to send status messages to any QRadar Console or any Event Processor or Event Collector in your deployment. Set the value to **Disabled** to send only a heartbeat without status messages. Set the value to **None** if you don't want to send a heartbeat or status messages. |

| *Table 13. WinCollect stand-alone setup type installation wizard parameters* | |
|---|---|
| **Parameter** | **Description** |
| **Create Log Source** | If this check box is selected, you must provide information about the log source and the target destination. |
| **Log Source Name** | The name can be a maximum length of 255 characters. |
| **Log Source Identifier** | Identifies the device that the WinCollect agent polls. This field must use the hostname, IP address, or FQDN of the Windows host that the log source gathers events from. |
| **Event Logs** | The Windows logs that you want the log source to collect events from and send to QRadar. |

| *Table 13. WinCollect stand-alone setup type installation wizard parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Destination Name** | Identifies where WinCollect events are sent. |
| **Hostname / IP** | The host name or IP address for the destination. |
| **Port** | The port that WinCollect uses when it communicates with the destination. |
| **Protocol** | **TCP** or **UDP** |
| **Machine poll interval (msec)** | The polling interval that determines the number of milliseconds between queries to the Windows host.<br>The minimum polling interval is 300 milliseconds. The default is 3000 milliseconds or 3 seconds. |
| **Event Rate Tuning Profile** | Select the tuning profile:<br>• Default (Endpoint): 100/150<br>This setting is suitable for Windows endpoints that are running a non-Server OS.<br>• Typical Server: 500/750<br>This setting is suitable for most Windows Server endpoints.<br>• High Event Rate Server: 1250/1875<br>This setting is suitable for all Windows endpoints and is ideal for Domain Controllers and other potentially high EPS endpoints.<br>For more information, see IBM Support (http://www-01.ibm.com/support/docview.wss?uid=swg21672193). |
| **Default Status Server Address** | The IP address Destination where status messages from the WinCollect agent are sent. |
| **Syslog Status Server (if different from default)** | An alternative destination to send WinCollect status messages to, such as the heartbeat, if required. Set the value to an IP address to send status messages to any QRadar Console or any Event Processor or Event Collector in your deployment. Set the value to **Disabled** to send only a heartbeat without status messages. Set the value to **None** if you don't want to send a heartbeat or status messages.<br>For QRadar on Cloud deployments, use the Data gateway. |
| **Heartbeat Interval (msecs)** | The frequency that heartbeat status messages are sent. In WinCollect 7.2.8, it is displayed in milliseconds. In WinCollect 7.2.9 and later, it is displayed in minutes. |

| Table 13. WinCollect stand-alone setup type installation wizard parameters (continued) | |
|---|---|
| **Parameter** | **Description** |
| **Log Monitor Socket Type** | Protocol to be used to send heartbeat and status messages.<br>**Note:** This option is only available in stand-alone WinCollect deployments. Availability for managed agents is planned in a later release of QRadar. |

The **Command Line (will be saved in config\cmdLine.txt)** field displays a command line from the configuration that you completed. You can use this command for silent, or unattended installations. For more information, see "Installing a WinCollect agent from the command prompt" on page 27.

# Installing a WinCollect agent from the command prompt

For unattended installations, you can install the WinCollect agent from the command prompt. Use the silent installation option to deploy WinCollect agents simultaneously to multiple remote systems.

## About this task

The WinCollect installer uses the following command options:

| Table 14. Silent installation options for WinCollect agents | |
|---|---|
| **Option** | **Valid entries and description** |
| `/qn` | Runs the WinCollect agent installation in silent mode. |
| `INSTALLDIR` | The installation location for WinCollect.<br>If the installation directory contains spaces, add a backslash before the quotation marks.<br>**Example:** `INSTALLDIR=\"C:\Program Files\IBM\WinCollect\"` |
| `AUTHTOKEN=token` | For managed WinCollect agents only. Uses the previously configured Authorization Token from QRadar to authorize the managed agent.<br>**Example:** `AUTH_TOKEN=af111ff6-4f30-11eb-11fb-1fc1 17711111` |
| `FULLCONSOLEADDRESS=host_address` | The IP address, host name, or FQDN of the QRadar Console, Event processor, or Event Collector that manages the agent.<br>**Examples:**<br>• `FULLCONSOLEADDRESS=192.0.2.0`<br>• `FULLCONSOLEADDRESS=EPqradar`<br>• `FULLCONSOLEADDRESS=EPqradar.myhost.com` |

*Table 14. Silent installation options for WinCollect agents (continued)*

| Option | Valid entries and description |
|---|---|
| HOSTNAME=host name | The **Hostname** field is used to assign a name to the WinCollect agent. The values that are used in this field can be an identifiable name, hostname, or IP address. In most cases, administrators can use HOSTNAME=%COMPUTERNAME% to auto populate this field.<br>**Example:** HOSTNAME="windows-%computername%"" HOSTNAME=WindowsSrv1 HOSTNAME=%COMPUTERNAME%<br>The IP address or host name of the WinCollect agent host cannot contain the "at" sign, @. |
| STATUSSERVER | An alternative destination to send WinCollect status messages to, such as the heartbeat, if required. Set the value to an IP address to send status messages to any QRadar Console or any Event Processor or Event Collector in your deployment. Set the value to Disabled to send only a heartbeat without status messages. Set the value to None if you don't want to send a heartbeat or status messages. |
| LOG_SOURCE_AUTO_CREATION_ENABLED | Required, True or False<br>If you enable this option, you must configure the log source parameters.<br>QRadar systems must be updated to V7.2.1 Patch 1 or later. |
| LOG_SOURCE_AUTO_CREATION_ PARAMETERS | Ensure that each parameter uses the format: Parameter_Name=value.<br>The parameters are separated with ampersands, &.<br>Your QRadar system must be updated to V7.2.1 Patch 1 or later. |
| LOG_MONITOR_SOCKET_TYPE=TCP | This parameter sets the protocol that is used by heartbeat and status messages to be sent by using TCP. The default protocol is UDP.<br>**Note:** This option is only available in stand-alone WinCollect deployments. Availability for managed agents is planned in a later release of QRadar. |
| Component1.Action | create<br>Creates a new windows event log source during the installation. |
| Component1.LogSourceIdentifier | The IP address or host name of the system where the agent is installed. |

*Table 14. Silent installation options for WinCollect agents (continued)*

| Option | Valid entries and description |
|---|---|
| `Component1.Destination.Name` | The destination name is an alphanumeric value that is used to specify where a WinCollect log source sends event data. This value must be a QRadar appliance capable of receiving event data, such as an Event Processor, Event Collector, or QRadar Console.<br>**Important:** In managed deployments, the destination must be an "internal destination," and the name must exist in the QRadar user interface before the installation. Otherwise, the log source configuration parameters are discarded and no log sources are automatically created.<br>**Internal Destination**<br>Managed hosts with an event processor component<br>**External Destination**<br>Destination that you configured as the WinCollect destination and is not known to the Console as a Managed Host |
| `Component1.Dest.Hostname`<br>(Stand alone deployments only) | The IP address or host name where you send WinCollect events. |
| `Component1.Dest.Port`<br>(Stand alone deployments only) | The port that WinCollect uses when it communicates with the destination. |
| `Component1.Dest.Protocol`<br>(Stand alone deployments only) | `TCP` or `UDP` |
| `Component1.Dest.MaxPayloadSize`<br>(Stand alone deployments only) | Maximum payload size sent to the destination (Default values are 1020 UDP and 32000 TCP). |
| `Component1.Log.Security` | Required, `True` or `False`<br>The Windows Security log contains events that are defined in the audit policies for the object. |
| `Component1.Log.System` | Required, `True` or `False`<br>The Windows System logs can contain information about device changes, device drivers, system changes, events, and operations provided by the operating system. |
| `Component1.Log.Application` | Required, `True` or `False`<br>The Windows Application logs contain events that are triggered by software applications instead of the operating system. The logs can contain errors, information, and warning events. |

*Table 14. Silent installation options for WinCollect agents (continued)*

| Option | Valid entries and description |
|---|---|
| `Component1.Log.DNS+Server` | Required, `True` or `False`<br>The Windows DNS Server log contains DNS events. |
| `Component1.Log.File+Replication+Service` | Required, `True` or `False`<br>The Windows File Replication Service log contains events about changed files that are replicated on the system. |
| `Component1.Log.Directory+Service` | Required, `True` or `False`<br>The Windows Directory Service log contains events that are written by the active directory. |
| `Component1.RemoteMachinePollInterval` | The polling interval that determines the number of milliseconds between queries to the Windows host.<br>The minimum polling interval is 300 milliseconds. The default is 3000 milliseconds or 3 seconds. |
| `Component1.EventRateTuningProfile`<br>(Managed deployments only) | Select one of the following tuning profiles:<br>• Default+(Endpoint)<br>• Typical+Server<br>• High+Event+Rate+Server<br>For more information, see IBM Support (http://www-01.ibm.com/support/docview.wss?uid=swg21672193). |
| `Component1.MaxLogsToProcessPerPass`<br>(Stand alone deployments only) | Not required.<br>The maximum number of logs (in binary form) that the algorithm attempts to acquire in one pass, if remaining retrievable events exist.<br>**Example:**<br>`Component1.MaxLogsToProcessPerPass=400`<br>**Important:** Use this parameter to improve performance for event collection, however, this parameter can also increase processor usage. For more information, see IBM Support (http://www-01.ibm.com/support/docview.wss?uid=swg21672193). |

| *Table 14. Silent installation options for WinCollect agents (continued)* | |
|---|---|
| **Option** | **Valid entries and description** |
| `Component1.MinLogsToProcessPerPass`<br>(Stand alone deployments only) | Not required.<br>The minimum number of logs (in binary form) that the algorithm attempts to read in one pass, if remaining retrievable events exist.<br>**Example:**<br>`Component1.MinLogsToProcessPerPass=200`<br>**Important:** You can use this parameter to improve performance for event collection, but this parameter can also increase processor usage. For more information, see IBM Support (http://www-01.ibm.com/support/docview.wss?uid=swg21672193). |
| `Component1.CoalesceEvents` | Not required.<br>Increases the QRadar event count when the same event occurs multiple times within a short time interval. Coalesced events provide a way to view and determine the frequency with which a single event type occurs on the Log Activity tab. When this option is disabled, events are viewed individually and events are not bundled. New and automatically discovered log sources inherit the value from the System Settings configuration on the Console. |
| `Component1.StoreEventPayload` | Not required.<br>Specifies that QRadar event payloads are to be stored. |
| `Component1.Secondary` | Not required.<br>Specifies the IP address or Hostname of the Secondary destination that the Agent sends events to if the Primary destination is unreachable and the failover time has elapsed. |
| `Component1.Failover` | Not required.<br>Specifies the failover time in seconds. If the primary destination can't be reached, the Agent starts sending events to the Secondary destination. |

**Attention:** You need to run the command prompt as an administrative user.

## Procedure

1. Download the WinCollect agent setup file from the IBM website (www.ibm.com/support).
2. On the Windows host, open a command prompt by using **Run as Administrator**.

**Important:** In managed deployments, the destination name that is used during automatic log source creation must exist before the command-line installation runs. Verify the destination name in the QRadar user interface before you start the installation.

3. Type the following command:

```
wincollect-<Version_number>.x64.exe /s /v" /qn
INSTALLDIR=<"C:\IBM\WinCollect">
AUTHTOKEN=<token> FULLCONSOLEADDRESS=<host_address>
HOSTNAME=<hostname> LOG_SOURCE_AUTO_CREATION=<true|false>
LOG_SOURCE_AUTO_CREATION_PARAMETERS=<"parameters""">
```

The following example shows a silent installation for a Stand alone WinCollect agent.

**Important:** This example contains line breaks for formatting. The actual command is a single line.

```
wincollect-<version_number>.x86.exe /s /v"/qn INSTALLDIR=\"C:\Program Files
\IBM\WinCollect\" HEARTBEAT_INTERVAL=6000 LOG_SOURCE_AUTO_CREATION_ENABLED=
True LOG_SOURCE_AUTO_CREATION_PARAMETERS=""Component1.AgentDevice=
DeviceWindowsLog&Component1.Action=create&Component1.LogSourceName=
%COMPUTERNAME%-1&Component1.LogSourceIdentifier=
<ip_address>&Component1.Dest.Name=QRadar&Component1
.Dest.Hostname=<ip_address>&Component1.Dest.Port=
514&Component1.Dest.Protocol=TCP&Component1.Log.Security=true&Component1
.Log.System=true&Component1.Log.Application=true
&Component1.Log.DNS+Server=false&Component1.Log.File+Replication+
Service=false&Component1.Log.Directory+Service=false&Component1.
RemoteMachinePollInterval=3000&Component1.EventRateTuningProfile=High+
Event+Rate+Server&Component1.MinLogs
ToProcessPerPass=1250&Component1.MaxLogsToProcessPerPass=1875"""
```

The following example shows a silent installation for a managed WinCollect agent.

**Important:** This example contains line breaks for formatting. The actual command is a single line.

```
 wincollect-<version_number>.x86.exe /s /v"/qn INSTALLDIR=\"C:\Program Files
\IBM\WinCollect\" AUTHTOKEN=1111111-aaaa-1111-aaaa-11111111
FULLCONSOLEADDRESS=<ip_address:port> HOSTNAME=%COMPUTERNAME%
 LOG_SOURCE_AUTO_CREATION_ENABLED=True LOG_SOURCE_AUTO_CREATION_PARAMETERS
=""Component1.AgentDevice=DeviceWindowsLog&Component1.Action=create
&Component1.LogSourceName=%COMPUTERNAME%&Component1.LogSourceIdentifier=
%COMPUTERNAME%&Component1.Log.Security=true&Component1.Log.System=false
&Component1.Log.Application=false&Component1.Log.DNS+Server=false
&Component1.Log.File+Replication+Service=false&Component1.Log.
Directory+Service=false&Component1.Destination.Name=Local&
Component1.RemoteMachinePollInterval=3000&Component1.EventRate
TuningProfile=High+Event+Rate+Server"""
```

4. Press Enter.

# Uninstalling a WinCollect agent from the command prompt

You can uninstall the WinCollect agent from the command prompt.

## Procedure

1. From the desktop, select **Start** > **Run**, type cmd, and click **OK**.

**Attention:** You need to run the command prompt as an administrative user.

2. If you want to remove all files, type the following command:

```
msiexec /x{1E933549-2407-4A06-8EC5-83313513AE4B} REMOVE_ALL_FILES=True /qn
```

3. If you want to remove just the WinCollect application, and not configuration files, stored events, and bookmarks, type the following command:

```
msiexec /x{1E933549-2407-4A06-8EC5-83313513AE4B} REMOVE_ALL_FILES=False /qn
```

4. Press Enter.

# Uninstalling a WinCollect agent from the Control Panel

You can uninstall the WinCollect agent from the Microsoft Windows Control Panel.

## Procedure

1. Click **Control Panel** > **Programs** > **Uninstall a program**.

   **Attention:** You need to start the control panel as an administrative user.
2. Highlight WinCollect in the program list, and click **Uninstall**.
3. If prompted by Windows, confirm that you want to remove WinCollect.

# Chapter 4. Configuring WinCollect agents after installation

In managed WinCollect deployments, you can use IBM Security QRadar for many agent configuration tasks. In stand-alone deployments, use the WinCollect Configuration Console to manage your WinCollect deployment.

Some WinCollect agent configurations must be performed on the Windows host where the agent is installed.

## Configuring managed WinCollect agents

After you install a managed WinCollect deployment, you manage your deployment by using IBM Security QRadar.

You can manage your WinCollect agents, destinations, and schedules. You can also manage configuration options for systems with restricted policies.

The WinCollect agent is responsible for communicating with the individual log sources, parsing events, and forwarding the event information to QRadar by using syslog.

After you install the WinCollect agent on your Windows host, wait for QRadar to automatically discover the WinCollect agent. The automatic discovery process typically takes a few minutes to complete.

**Note:** The registration request to the QRadar host might be blocked by firewalls in your network.

### Manually adding a WinCollect agent

If you delete your WinCollect agent, you can manually add it back. To reconnect to an existing WinCollect agent, the host name must exactly match the host name that you used before you deleted the agent.

When you delete a WinCollect agent, the IBM Security QRadar Console removes the agent from the agent list and disables all of the log sources that are managed by the deleted WinCollect agent.

WinCollect agents that were previously automatically discovered are not rediscovered in WinCollect. To add a deleted WinCollect agent back to the agent list in the QRadar, you must manually add the deleted agent.

For example, you delete a WinCollect agent that has a host identifier name VMRack1. You reinstall the agent and use the same host identifier name, VMRack1. The WinCollect agent does not automatically discover the WinCollect agent.

#### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click **Agents**.
4. Click **Add**.
5. Configure the parameters.

   The following table describes some of the parameters:

| *Table 15. WinCollect agent parameters* | |
|---|---|
| **Parameter** | **Description** |
| **Host Name** | Depending on the method that you used to install the WinCollect agent on the remote host, the value in the **Host Name** field must match one of the following values:<br>• **HOSTNAME** field in the WinCollect agent command-line configuration<br>• **Host Identifier** field in the WinCollect agent installer. |
| **Description** | Optional.<br>If you specified an IP address as the name of the WinCollect agent, add descriptive text to identify the WinCollect agent or the log sources the WinCollect agent is managing. |
| **Automatic Updates Enabled** | Controls whether configuration updates are sent to the WinCollect agent. |
| **Heart Beat Interval** | This option defines how often the WinCollect agent communicates its status to the Status Server. The interval ranges from 1 - 20 minutes. |
| **Configuration Poll Interval** | Defines how often the WinCollect agent polls the QRadar Configuration server for updated log source configuration information or agent software updates. The interval ranges from 1 minute to 20 minutes. |
| **Maximum TCP/UDP Payload** | The WinCollect agent uses whatever setting is selected in QRadar for the maximum TCP/UDP payload size. |

6. Click **Save**.
7. On the **Admin** tab, click **Deploy Changes**.

   The WinCollect agent is added to the agent list.

**Related tasks**

Deleting a WinCollect agent

When you delete a WinCollect agent, the IBM Security QRadar Console removes the agent from the agent list and disables all of the log sources that are managed by the deleted WinCollect agent.

# Deleting a WinCollect agent

When you delete a WinCollect agent, the IBM Security QRadar Console removes the agent from the agent list and disables all of the log sources that are managed by the deleted WinCollect agent.

## Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Select the agents that you want to delete and click **Delete**.
5. Click **Save**.

6. On the **Admin** tab, click **Deploy Changes**.

   **Tip:** To delete multiple WinCollect agents, press Ctrl to select multiple agents, and then click **Delete**.

**Related tasks**

Manually adding a WinCollect agent

# WinCollect destinations

WinCollect destinations define the parameters for how the WinCollect agent forwards events to the Event Collector or IBM Security QRadar Console.

## Adding a destination

To assign where WinCollect agents in your deployment forward their events, you can create destinations for your WinCollect deployment.

### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Click **Destinations** and then click **Add**.
5. Configure the parameters.

   The following table describes some of the parameters

*Table 16. Destination parameters*

| Parameter | Description |
|---|---|
| **Name** | Used on the agent side for log source creation.<br>**Important:** The destination name is used during automatic log source creation and must exist before the installation runs. Verify the destination name in QRadar before starting the installation. |
| **Hostname** | The host name or IP address of the destination IBM QRadar appliance. |
| **Port** | IBM Security QRadar receives events from WinCollect agents on UDP or TCP on port 514.<br>For TLS protocol, the default port is 6514. |
| **Protocol** | The communication channel between IBM Security QRadar and WinCollect agents. Select **UDP**, or **TCP**, or **TCP/TLS (Encrypted)**. |
| **Certificate** | The TLS certificate of the destination device.<br>Copy the certificate from `/opt/qradar/conf/trusted_certificates/syslog-tls.cert` on the destination device and paste in the **Certificate** field.<br>**Note:** The **Certificate** field displays when **TCP/TLS (Encrypted)** is selected from the **Protocol** list. |

| *Table 16. Destination parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Throttle (events per second)** | Defines a limit to the number of events that the WinCollect agent can send each second. |
| **Schedule Mode** | If you select the **Forward Events** option, the WinCollect agent forwards events within a user-defined schedule. When the events are not being forwarded, they are stored until the schedule runs again.<br>If you select the **Store Events** option, the WinCollect agent stores events to disk only within a user-defined schedule and then forwards events to the destination as specified. |

6. Click **Save**.

**Related tasks**

Adding a secondary destination

You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

Deleting a destination from WinCollect

Scheduling event forwarding and event storage for WinCollect agent

## Adding a secondary destination

You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

### About this task

**Note:** Adding a secondary destination is available in IBM QRadar 7.4.3 and later.

Use the following procedure to add a QRadar host as a secondary destination to an existing primary destination. For more information about adding a secondary destination during the installation process, see “Adding a destination” on page 37.

**Note:** To specify a secondary destination, you must select **TCP**.

### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click **WinCollect** > **Destinations**.
4. Select a destination and click **Edit**.
5. Select the TCP **Protocol**.
6. Enter the hostname or IP address of the IBM QRadar appliance you want to use as a **Secondary Destination**.
7. In the **Secondary Failover (seconds)** field, enter the number of seconds that the primary destination must be unreachable before the agent begins sending events to the secondary destination.
8. Click **Save**.

**Related tasks**

Adding a destination

To assign where WinCollect agents in your deployment forward their events, you can create destinations for your WinCollect deployment.

Deleting a destination from WinCollect
Scheduling event forwarding and event storage for WinCollect agent

## Deleting a destination from WinCollect

If you delete a destination, the event forwarding parameters are removed from the WinCollect agent.

Destinations are a global parameter. If you delete a destination when log sources are assigned to the destination, the WinCollect agent cannot forward events. Event collection is stopped for a log source when an existing destination is deleted. Events on disk that were not processed are discarded when the destination is deleted.

### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Click **Destinations**.
5. Select the destination that you want to delete and click **Delete**.

**Related tasks**

Adding a destination
To assign where WinCollect agents in your deployment forward their events, you can create destinations for your WinCollect deployment.

Adding a secondary destination
You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

Scheduling event forwarding and event storage for WinCollect agent

## Scheduling event forwarding and event storage for WinCollect agent

Use a schedule to manage when WinCollect agents forward or store events to disk in your deployment.

Schedules are not required. If a schedule does not exist, the WinCollect agent automatically forwards events and stores them only when network limitations cause delays.

You can create schedules for your WinCollect deployment to assign when the WinCollect agents in your deployment forward their events. Events that are unable to be sent during the schedule are automatically queued for the next available interval.

### Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Click **Schedules**.
5. Click **Add** and then click **Next**.
6. Configure the parameters, and select a check box for each day of the week that you want included in the schedule.
7. Click **Next**.
8. To add a destination to the schedule, from the **Available Destinations** list, select a destination and click the selection symbol, >.
9. Click **Next** and then click **Finish**.

**Related tasks**

Adding a destination
To assign where WinCollect agents in your deployment forward their events, you can create destinations for your WinCollect deployment.

Adding a secondary destination
You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

Deleting a destination from WinCollect

# Adding custom entries to WinCollect status messages

You can add custom information to the WinCollect Agent status messages.

## Procedure

1. In the `wincollect/config` directory of the Windows host that you want to identify in LEEF logs, create a file that is called `heartbeat_custom.props`.

   **Important:** You can create, update, or delete this file while your WinCollect deployment is running. Updates to the file are available in logs on the next heartbeat.
2. Enter the custom information in the `heartbeat_custom.props` file in the following format, with one entry on each line:

   ```
   keyword=value
   ```

   **Example:**

   ```
   department=Accounting
   ```

   ```
   group=AC105
   ```

   The log output with the example keywords and values looks like the following example:

   ```
   <13>Jul 22 15:02:48 DESKTOP-0F0QKN3 LEEF:1.0|IBM|
   WinCollect|<version_number>.9999|2|src=DESKTOP-0F0QKN3
   os=Windows 10(Build 10240 64-bit)dst= sev=3 log=Code.SSLConfigServerConnection
   department=Accounting group=AC105 msg=ApplicationHeartbeat
   ```

   **Important:**

   - The `heartbeat_custom.props` must not exceed 10 KB.
   - Custom keyword entries must be alpha-numeric and contain no spaces.
   - Custom entries can't contain reserved keywords, such as `src, os, dst, sev, log, msg`.
   - Custom values can't contain special characters, such as `= | [ ] { } < > / \ ' " .`
   - Multiple white spaces in custom values are reduced to a single space.

# Forwarded Events Identifier

If you enable a log source to collect forwarded events using Windows event subscriptions, you can specify the event source displayed for each event. Configure the Forwarded Events Identifier in the log source that collects forwarded events.

There are 3 options for setting the Forwarded Events Identifier:

**Source**
This is the default option. Forwarded events are identified by the IP address of the computer that generated the events.

**WEC**
Forwarded events are identified by the name of the WinCollect agent that collects them. All events collected by the Agent are grouped together with a single source identifier.

**Other**
You can choose a custom identifier as the source for the events. All events collected by the Agent are grouped together with this identifier.

**Tip:** Custom identifiers cannot contain spaces.

# Configuring stand-alone WinCollect agents with the Configuration Console

In stand-alone deployments, use the WinCollect Configuration Console to manage your WinCollect deployment.

Some WinCollect agent configurations must be performed on the Windows host where the agent is installed.

## Creating a WinCollect credential

Create a credential that contains login information. WinCollect uses the credential information to log into devices and collect logs.

### Procedure

1. Expand the **Global Configuration** parameter and right-click **Security Manager**.
2. Select **Add New Credential**.
3. In the **New Credential Name** box, add a name for the new credential and click **OK**.
4. Click the new credential under **Security Manager** to open the **Basic Configurations** window for the credential.
5. Enter the required properties for the new credential.
6. Click **Deploy Changes** under **Actions**.

## Adding a destination to the WinCollect Configuration Console

Add an IBM QRadar instance as a destination for WinCollect data.

### Procedure

1. In the WinCollect Configuration Console, expand the **Destinations** parameter.
2. Right-click the **Syslog TCP** or **Syslog UDP** parameter, depending upon which destination type you want to add, and click **Add New Destination**.

   **Note:** If you want to specify a secondary destination, you must select the TCP **Protocol**.
3. In the **New Destination Name** box, add a name for the destination. Click **OK**.

   **Important:** It is helpful to provide a destination name that includes the IP address, such as `QRadarEP1_198.x.x.x`. If you have to edit the log source and change a destination in the future, you can determine the IP address for the destination.
4. Expand **Syslog TCP** or **Syslog UDP**, and select the destination that you added to view the **Properties** window.
5. Define the **Name**, **Hostname**, **Port**, and **Throttle** for the new destination.
6. If you have Data Sync and want to add a **Secondary Destination** to receive events if the primary destination fails, add the IP address or hostname.

7. If you added a **Secondary Destination**, enter the number of seconds that the primary destination must be unreachable before the agent begins sending events to the secondary destination in the **Secondary Failover (seconds)** field.
8. Click **Deploy Changes** under **Actions**.

   **Note:** Stand-alone deployments of WinCollect 7.3.0 and later support adding a secondary destination.

## Configuring a destination with TLS in the WinCollect Configuration Console

You can encrypt syslog traffic to be sent to QRadar by configuring the WinCollect destination to use a Transport Layer Security (TLS) certificate.

### Procedure

1. In the WinCollect Configuration Console, expand the **Destinations** parameter.
2. Right-click the **Syslog TCP**, and click **Add New Destination**.
3. In the **New Destination Name** field, add a name for the destination, and click **OK**.

   **Tip:** Use a destination name that includes the IP address, such as "*<Managed_Host>*_1.2.3.4". If you need to edit the log source and change a destination in the future, this destination name helps you determine the IP address for the destination.
4. Expand **Syslog TCP**, and select the destination that you added in step 3 to view the **Properties** window.
5. Define the **Name** and **Hostname**.
6. Change the **Port** to 6514, and set the **Throttle** rate.
7. Copy and paste the TLS certificate for the new destination in the **Certificate** field.

   **Note:** Make sure that you include the "`-----BEGIN CERTIFICATE-----`" and the "`-----END CERTIFICATE-----`" when you copy the TLS certificate.
8. Click **Deploy Changes** under the **Actions** pane.

## Adding a device to the WinCollect Configuration Console

Add the devices that WinCollect monitors to the WinCollect Configuration Console.

### Procedure

1. Under **Devices**, right-click the device type that matches the device you want to add and select **Add New Device**.
2. In the **Add New Device** box, enter a name for the destination device.
3. In the **Basic Configurations** window, complete the parameters for the new destination device.

   **Important:** On the **Basic Configurations** page of the Microsoft Windows Event log device type, you can set a global Default Event Log Poll Protocol. The default value is **MSEVEN6**.

   To configure a single Microsoft Windows Event Log device to use the global Default Event Log Poll Protocol, select **default** from the **Basic Configurations** page of the device. Otherwise, select **MSEVEN6** or **MSEVEN** to override the global Default Event Log Poll Protocol.

   The **MSEVEN6** is a Microsoft event protocol that collects more information from an event log, such as the task, keyword, and opcode. It also provides a better message formatting.
4. Click **Deploy Changes** under **Actions**.

## Sending encrypted events to QRadar

Configure a log source in stand-alone deployments of WinCollect to send encrypted events to IBM QRadar with TLS syslog. TLS Syslog is only supported in managed WinCollect deployments in QRadar versions 7.3.1 and later.

### Before you begin

In QRadar, configure a Universal DSM that uses the TLS Syslog protocol. For more information, see the *IBM Security QRadar Log Sources User Guide*.

The uDSM opens a port and provides the certificate that is necessary for communicating by using TLS. If you delete the uDSM, TLS communication stops.

### Procedure

1. Use SSH to log in to QRadar as the root user.
2. Copy the certificate, including `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` from `/opt/qradar/conf/trusted_certificates/syslog-tls.cert` to a temporary location. You will paste this certificate into the WinCollect Configuration Console.
3. In the WinCollect Configuration Console, expand **Destinations**, and click **Add Destination**.
4. In the **New Destination Name** box, add a name for the destination and then click **OK**.
5. Select the new destination and enter the IP address of the target QRadar appliance in the **Hostname** field.
6. Type 6514 in the **Port** field.
7. Type the events per second (EPS) rate for your deployment in the **Throttle** field.
8. Paste the certificate that you copied from QRadar into the **Certificate** field.
9. Click **Deploy Changes** under **Actions**.

## Increasing UDP payload size

You can increase the payload size for UDP syslog destinations in the Agent Configuration file.

### About this task

The default payload size for UDP destination packages is 1,024 bytes. You can increase the payload size for a stand-alone WinCollect agent by adding a parameter in the Agent Configuration file.

**Important:** After you change the payload size for the WinCollect agent, you must increase the maximum UDP payload size in QRadar.

### Procedure

1. Open the Agent Configuration XML file.

   The default path to this file is `WinCollect\config\AgentConfig.xml`.
2. Add the following parameter to the `UDPSendStage` module:

   ```
   <Parameter name="MaxPayloadSize" value="<desired value>" />
   ```

   Example of the module:

   ```
   <Module order="4" service_name="UDPSendStage">
           <Environment>
              <Parameter value="<Destination IP>" name="TargetAddress"/>
              <Parameter value="514" name="TargetPort"/>
              <Parameter name="MaxPayloadSize" value="4096"/>
           </Environment>
       </Module>
   ```
3. Save the file, and restart the WinCollect agent.

### What to do next

After you change the payload size for the WinCollect agent, you must increase the maximum UDP payload size in QRadar. For more information on increasing payload size in QRadar, see TCP and UDP Syslog maximum payload message length for QRadar appliances.

## Include milliseconds in Event Log timestamp

In a stand-alone WinCollect deployment, you can include milliseconds in the timestamp for Event Logs.

**Note:** This option is only compatible in a stand-alone WinCollect deployment that uses the MSEVEN6 protocol. It is not supported by the MSEVEN protocol

The **TimeGenerated** and **TimeWritten** payload fields in the Event Logs use seconds by default. You can set the **Timestamp Properties** to use milliseconds in the **Microsoft Windows Event Log Properties** node of the **WinCollect Configuration Console**.

**Important:** This is an Agent-level change that is set for all log sources.

Alternatively, you can change the property as part of the command line installation, using this parameter: `&Component1.TimestampFormat=Milliseconds`. You can also use a template to change the attribute in the `AgentConfig.xml` file. For more information about using templates, see "Changing configuration with templates in a stand-alone deployment" on page 45.

## Collecting local Windows logs

This use case scenario describes the settings required to collect logs from the host where the WinCollect Configuration Console is installed, and send them to IBM QRadar.

### Procedure

1. Install the WinCollect Configuration Console on the host on which that you want to collect windows logs. Download the patch from IBM Support (www.ibm.com/support/fixcentral).
2. Create a destination for the QRadar instance where you want to send WinCollect information. See "Adding a destination to the WinCollect Configuration Console" on page 41.
3. Configure the local Microsoft event log device that is monitored. See "Adding a device to the WinCollect Configuration Console" on page 42.

   **Important:** In the **Device Address** field, type the IP address or hostname of the local Windows system that you want to poll for events.
4. Click **Deploy Changes** under **Actions**.

## Collecting remote Windows logs

This use case scenario describes the settings that are required in the WinCollect Configuration Console to collect windows logs from hosts that do not have WinCollect software installed, and send the logs to IBM QRadar.

### About this task

**Note:** WinCollect does not support reverting Citrix Virtual Machines that are polled remotely.

### Procedure

1. Install the WinCollect Configuration Console on the windows machine that collects the log information. Download the patch from IBM Support (www.ibm.com/support/fixcentral).
2. Create a credential to use when you log in to remote hosts. See "Creating a WinCollect credential" on page 41.
3. Create the QRadar destination where Windows events are sent. See "Adding a destination to the WinCollect Configuration Console" on page 41.

4. Configure the devices that are monitored. See “Adding a device to the WinCollect Configuration Console” on page 42.

   **Important:** In the **Device Address** field, type the IP address or hostname of the remote Windows system that you want to poll for events.
5. Click **Deploy Changes** under **Actions**.

# Changing configuration with templates in a stand-alone deployment

Supported Version: WinCollect 7.2.8+ stand-alone only.
With templating, you can change the Agent configuration without making manual or scripted edits to the `AgentConfig.xml` file.

When you copy a template to the WinCollect patch directory, the Agent replaces the existing configuration with the contents of the template. Before the Agent applies the changes from the template, it makes a backup of the current configuration in the `patchcheckpoint` directory. After the changes are applied, the Agent restarts and uses the new configuration.

Four sample templates are installed with WinCollect V7.2.8 and later. They are stored in the `\IBM\WinCollect\templates` directory.

- `tmplt_AgentCore.xml`
- `tmplt_DestinationManager.xml`
- `tmplt_DeviceWindowsLog.xml`
- `tmplt_PayloadRouter.xml`

**Note:** These templates are examples only. All Agent configuration service modules are supported, so that you can create your own templates.

The following use cases are examples of how you can use templates to change Agent configurations.

## Use Case 1: Change heartbeat interval

You want to change the heartbeat interval from 5 minutes to 1 hour on all deployed systems. Previously, this required manual or scripted changes to the `agentconfig.xml` file and a WinCollect service restart. With templates, you can change this interval by performing the following steps.

### Procedure

1. Locate the `tmplt_AgentCore.xml` template in the `\IBM \WinCollect\templates` directory. This service contains the Heartbeat Interval configuration.
2. Make a copy of the template and name it `service_AgentCore.xml`.
3. Change the value of the `HeartbeatInterval` parameter to 3,600,000 milliseconds (1 hour).

```
<Service classification="Static" type="Service" version="7.2.8" module="AgentCore"
name="AgentCore">
        <Environment>
            <Parameter name="HeartbeatInterval" value="3600000"/>
            <Parameter name="ConfigurationCheckInterval" value="300000"/>
            <Parameter name="Enabled" value="true"/>
            <Parameter name="Deleted" value="false"/>
        </Environment>
    </Service>
```

4. Move the `service_AgentCore.xml` file to the `\IBM\WinCollect\patch` directory.

   After a few seconds, the file disappears and the agent restarts. The old `agentconfig.xml` file is moved to the backup directory (`patch_checkpoint_`*xxxx*).

## Use Case 2: Modify event data storage configuration

### About this task

You want to change the location and capacity of the event data that is stored in the `\programdata\WinCollect` file. You want to store the event data in `C:\WinCollect\Data` and change the capacity to 20 GB. There is no default template for this change, but you can easily create one by using information in the `agentconfig.xml` file. The following sample shows the existing service:

```
<Service classification="Service" type="Service" version="7.2.8" module="WinCollectCommon"
 name="DiskManager">
        <Environment>
            <Parameter name="BasePath" value="%ALLUSERSPROFILE%\WinCollect\Data"/>
            <Parameter name="Capacity" value="6144"/>
        </Environment>
    </Service>
```

**Note:** `%ALLUSERSPROFILE%` is an environment variable. The default value is `C:\ProgramData`. You want to change this value to `C:\WinCollect\Data`.

### Procedure

1. Create an XML file named `service_DiskManager.xml` with the following contents:

```
<Service classification="Service" type="Service" version="7.2.8" module="WinCollectCommon"
 name="DiskManager">
        <Environment>
            <Parameter name="BasePath" value="c:\ibm\WinCollect\Data"/>
            <Parameter name="Capacity" value="20480"/>
        </Environment>
    </Service>
```

2. Move the file to the `\IBM\WinCollect\patch` directory.

   After a few seconds, the file disappears and the agent restarts. Data is now written to the new directory.

## Use Case 3: Send TCP instead of UDP

You want to send Syslog data to QRadar over TCP rather than UDP. You must specify this option in the Destination Manager.

### Procedure

1. Locate the `tmplt_DestinationManager.xml` template in the `\IBM \WinCollect\templates` directory.
2. Make a copy of the template and name it `service_DestinationManager.xml`.
3. In `<Module order="4" service_name="UDPSendStage">`, change the `service_name` parameter to TCPSendStage.

```
Service version="7.2.8" classification="Service" type="Service" module="WinCollectPlugin"
 name="DestinationManager">
    <Environment/>
    <InstanceData>
        <Instance name="QRadar">
            <Environment/>
            <Module order="1" service_name="StoreAndForwardStage">
                <Environment>
                    <Parameter name="DataChunkPeriod" value="10"/>
                    <Parameter name="DataProcessingPeriod" value="500000"/>
                    <Parameter name="QueueLowWaterMark" value="750000"/>
                    <Parameter name="QueueHighWaterMark" value="1000000"/>
                    <Parameter name="Schedule.Enable" value="true"/>
                    <Parameter name="Schedule.Invert" value="false"/>
                    <Parameter name="Socket.KeepAlive.Enabled" value="true"/>
                    <Parameter name="Socket.KeepAlive.Time" value="30000"/>
                    <Parameter name="Socket.KeepAlive.Interval" value="4000"/>
                </Environment>
            </Module>
```

```
            <Module order="2" service_name="SimpleEventThrottle">
                <Environment>
                    <Parameter name="EventThrottleInEPS" value="5000"/>
                </Environment>
            </Module>
            <Module order="3" service_name="SyslogHeaderStage">
                <Environment/>
            </Module>
            <Module order="4" service_name="TCPSendStage">
                <Environment>
                    <Parameter name="TargetAddress" value="172.18.X.X"/>
                    <Parameter name="TargetPort" value="514"/>
                </Environment>
            </Module>
        </Instance>
    </InstanceData>
</Service>
```

4. Move the file to the `\IBM\WinCollect\patch` directory.

   After a few seconds, the file disappears and the agent restarts. The old `agentconfig.xml` file is moved to the backup directory (`patch_checkpoint_`*xxxx*).

## Use Case 4: Add NSA filtering to an existing log source

You want to add NSA filtering to an existing log source. You can change this attribute by using the `tmplt_DeviceWindowsLog.xml` template.

### Procedure

1. Locate the `tmplt_DeviceWindowsLog.xml` template.
2. Make a copy of the template and name it `service_DeviceWindowsLog.xml`.
3. Open `AgentConfig.xml` and locate the `log source` contained in the module `DeviceWindowsLog`.
4. Copy the model and instance information and replace the contents in `service_DeviceWindowsLog.xml` with it.

   Existing log source example:

```
<Service version="7.2.8" classification="Service" type="DeviceType"
module="DeviceWindowsLog" name="DeviceWindowsLog">
    <Environment>
        <Parameter name="DeviceThreadPoolType" value="AdaptiveThreadPool"/>
        <Parameter name="AdaptiveThreadPool.ReaderThreadsMax" value="500"/>
        <Parameter name="AdaptiveThreadPool.ReaderThreadsMin" value="5"/>
        <Parameter name="AdaptiveThreadPool.ReaderBacklogSamplePeriodMillis" value="200"/>
        <Parameter name="MinEventMonitorThreads" value="5"/>
        <Parameter name="MaxEventMonitorThreads" value="250"/>
        <Parameter name="EventLogMonitor.RetryTimeoutMillis" value="60000"/>
        <Parameter name="DefaultThrottleTimeout" value="1500"/>
        <Parameter name="DefaultEventLogPollProtocol" value="MSEVEN6"/>
    </Environment>
    <InstanceData>
        <Instance enabled="true" name="EventLogLocal">
            <Environment>
                <Parameter name="DeviceAddress" value="DESKTOP"/>
                <Parameter name="RemoteMachine" value="DESKTOP"/>
                <Parameter name="Filter.DNS Server.Enabled" value="false"/>
                <Parameter name="EventTypeFilterFailureAudit" value="true"/>
                <Parameter name="EventLogPollProtocol" value="MSEVEN6"/>
                <Parameter name="Log.Security" value="true"/>
                <Parameter name="Filter.Application.Enabled" value="false"/>
                <Parameter name="ADLookup.Enabled" value="false"/>
                <Parameter name="ThrottleTimeout" value="1000"/>
                <Parameter name="Filter.DNS Server.Param" value=""/>
                <Parameter name="Filter.File Replication Service.Enabled" value="false"/>
                <Parameter name="Filter.Application.Type" value="No Filtering"/>
                <Parameter name="Filter.Directory Service.Param" value=""/>
                <Parameter name="Log.Application" value="true"/>
                <Parameter name="Filter.System.Type" value="No Filtering"/>
                <Parameter name="Filter.DNS Server.Type" value="No Filtering"/>
                <Parameter name="Filter.Application.Param" value=""/>
                <Parameter name="Filter.System.Param" value=""/>
                <Parameter name="Log.Directory Service" value="false"/>
                <Parameter name="ADLookup.DomainControllerName" value=""/>
                <Parameter name="Log.File Replication Service" value="false"/>
```

```
                <Parameter name="Filter.Directory Service.Enabled" value="false"/>
                <Parameter name="CustomQuery.Base64" value=""/>
                <Parameter name="Filter.Security.Param" value=""/>
                <Parameter name="EventRateTuningProfile" value="High Event Rate Server"/>
                <Parameter name="Local.System" value="true"/>
                <Parameter name="EventTypeFilterError" value="true"/>
                <Parameter name="EventTypeFilterWarn" value="true"/>
                <Parameter name="EventTypeFilterInfo" value="true"/>
                <Parameter name="Filter.File Replication Service.Param" value=""/>
                <Parameter name="Filter.File Replication Service.Type" value="No Filtering"/>
                <Parameter name="EventTypeFilterSuccessAudit" value="true"/>
                <Parameter name="Filter.Directory Service.Type" value="No Filtering"/>
                <Parameter name="Filter.Security.Type" value="No Filtering"/>
                <Parameter name="Application" value="None"/>
                <Parameter name="Log.System" value="true"/>
                <Parameter name="Log.ForwardedEvents" value="false"/>
                <Parameter name="Filter.Security.Enabled" value="false"/>
                <Parameter name="Filter.System.Enabled" value="false"/>
                <Parameter name="Log.DNS Server" value="false"/>
                <Parameter name="ADLookup.DNSDomainName" value=""/>
                <Parameter name="RemoteMachinePollInterval" value="3000"/>
                <Parameter name="MinLogsToProcessPerPass" value="1250"/>
                <Parameter name="MaxLogsToProcessPerPass" value="1825"/>
                <Parameter name="Login.Handle" value="0"/>
            </Environment>
        </Instance>
    </InstanceData>
</Service>
```

5. Modify the following lines with the bolded sample code:

```
<Parameter name="Filter.System.Type" value="NSAlist"/>
<Parameter name="Filter.System.Param" value=
"1,6,12,13,19,104,219,1001,1125,1126,1129,7000,7022,7023,7024,7026,7031,7032,7034,7045"/>
<Parameter name="Filter.System.Enabled" value="true"/>
```

6. Save the `service_DeviceWindowsLog.xml` file and move it to the `\IBM\WinCollect\patch` directory.

   After a few seconds, the file disappears and the agent restarts. The old `agentconfig.xml` file is moved to the backup directory (`patch_checkpoint_`*xxxx*). Updated log source example:

```
<Service version="7.2.8" classification="Service" type="DeviceType"
module="DeviceWindowsLog" name="DeviceWindowsLog">
    <Environment>
        <Parameter name="DeviceThreadPoolType" value="AdaptiveThreadPool"/>
        <Parameter name="AdaptiveThreadPool.ReaderThreadsMax" value="500"/>
        <Parameter name="AdaptiveThreadPool.ReaderThreadsMin" value="5"/>
        <Parameter name="AdaptiveThreadPool.ReaderBacklogSamplePeriodMillis" value="200"/>
        <Parameter name="MinEventMonitorThreads" value="5"/>
        <Parameter name="MaxEventMonitorThreads" value="250"/>
        <Parameter name="EventLogMonitor.RetryTimeoutMillis" value="60000"/>
        <Parameter name="DefaultThrottleTimeout" value="1500"/>
        <Parameter name="DefaultEventLogPollProtocol" value="MSEVEN6"/>
    </Environment>
    <InstanceData>
        <Instance enabled="true" name="EventLogLocal">
            <Environment>
                <Parameter name="DeviceAddress" value="DESKTOP"/>
                <Parameter name="RemoteMachine" value="DESKTOP"/>
                <Parameter name="Filter.DNS Server.Enabled" value="false"/>
                <Parameter name="EventTypeFilterFailureAudit" value="true"/>
                <Parameter name="EventLogPollProtocol" value="MSEVEN6"/>
                <Parameter name="Log.Security" value="true"/>
                <Parameter name="Filter.Application.Enabled" value="false"/>
                <Parameter name="ADLookup.Enabled" value="false"/>
                <Parameter name="ThrottleTimeout" value="1000"/>
                <Parameter name="Filter.DNS Server.Param" value=""/>
                <Parameter name="Filter.File Replication Service.Enabled" value="false"/>
                <Parameter name="Filter.Application.Type" value="No Filtering"/>
                <Parameter name="Filter.Directory Service.Param" value=""/>
                <Parameter name="Log.Application" value="true"/>
                <Parameter name="Filter.DNS Server.Type" value="No Filtering"/>
                <Parameter name="Filter.Application.Param" value=""/>
                <Parameter name="Filter.System.Type" value="NSAlist"/>
                <Parameter name="Filter.System.Param"
value="1,6,12,13,19,104,219,1001,1125,1126,1129,7000,7022,7023,7024,7026,7031,7032,7034,7045"
/>
                <Parameter name="Filter.System.Enabled" value="true"/>
```

```
                <Parameter name="Log.Directory Service" value="false"/>
                <Parameter name="ADLookup.DomainControllerName" value=""/>
                <Parameter name="Log.File Replication Service" value="false"/>
                <Parameter name="Filter.Directory Service.Enabled" value="false"/>
                <Parameter name="CustomQuery.Base64" value=""/>
                <Parameter name="Filter.Security.Param" value=""/>
                <Parameter name="EventRateTuningProfile" value="High Event Rate Server"/>
                <Parameter name="Local.System" value="true"/>
                <Parameter name="EventTypeFilterError" value="true"/>
                <Parameter name="EventTypeFilterWarn" value="true"/>
                <Parameter name="EventTypeFilterInfo" value="true"/>
                <Parameter name="Filter.File Replication Service.Param" value=""/>
                <Parameter name="Filter.File Replication Service.Type" value="No Filtering"/>
                <Parameter name="EventTypeFilterSuccessAudit" value="true"/>
                <Parameter name="Filter.Directory Service.Type" value="No Filtering"/>
                <Parameter name="Filter.Security.Type" value="No Filtering"/>
                <Parameter name="Application" value="None"/>
                <Parameter name="Log.System" value="true"/>
                <Parameter name="Log.ForwardedEvents" value="false"/>
                <Parameter name="Filter.Security.Enabled" value="false"/>
                <Parameter name="Log.DNS Server" value="false"/>
                <Parameter name="ADLookup.DNSDomainName" value=""/>
                <Parameter name="RemoteMachinePollInterval" value="3000"/>
                <Parameter name="MinLogsToProcessPerPass" value="1250"/>
                <Parameter name="MaxLogsToProcessPerPass" value="1825"/>
                <Parameter name="Login.Handle" value="0"/>
            </Environment>
        </Instance>
    </InstanceData>
</Service>
```

## Configuration options for systems with restricted policies for domain controller credentials

Users with appropriate remote access permissions might be able to collect events from remote systems without using domain administrator credentials. Depending on what information you collect, the user might need extra permissions. For example, a user might need to collect Security event logs remotely. Therefore, the user that is configured in the QRadar log source must have remote access to the Security event log from the server where the Agent is installed.

**Restriction:**

For remote collection, the WinCollect user must work with their Windows administrator to ensure access to the following items:

- Logs for security, system, and application events
- The remote registry
- Any directories that contain .dll or .exe files that contain message string information

With certain combinations of Windows operating system and group policies in place, alternative configurations might not be possible.

Remote collection inside or across a Windows domain might require domain administrator credentials to ensure that events can be collected. If your corporate policies restrict the use of domain administrator credentials, you might need to complete more configuration steps for your WinCollect deployment.

The following permissions and credentials are required for service accounts to access remote polling log sources that WinCollect supports.

| Permissions | Log Sources |
|---|---|
| The service account needs to be able to access the folder that the log file is in and open the file. | • Microsoft DHCP<br>• Microsoft Exchange Server<br>• DNS debug<br>• File Forwarder |

| Permissions | Log Sources |
|---|---|
| | • Microsoft IAS<br>• Microsoft IIS<br>• Microsoft ISA<br>• Juniper Steel-Belted Radius<br>• Microsoft SQL<br>• Net App Data ONTAP<br>• TLS |
| The log source user must be a member of the **Event Log Readers** group. If this group is not configured, then domain administrative privileges are usually required to poll a Windows event log across a domain. | Microsoft Windows Security Event Log |

When WinCollect agents collect events from the local host, the event collection service uses the Local System account credentials to collect and forward events. Local collection requires that you install a WinCollect agent on a host where local collection occurs.

## Changing WinCollect configuration from the command prompt

You can change the configuration of a WinCollect agent from the command prompt of the Windows host.

After the initial installation of a WinCollect agent on a Windows host, you can change the configuration by using the `installhelper.exe` file that is located in the *<WinCollect_installation_path>*`/bin`.

The following configuration parameters can be modified:

*Table 17. Modifiable configuration parameters*

| Parameter | Description |
|---|---|
| **Authentication Token** | Authorizes the WinCollect service, for example, `AUTH_TOKEN=af111ff6-4f30-11eb-11fb-1fc1 17711111` |
| **Password** | Update a password in the `AgentConfig.xml` configuration file. Specify the `Login.Handle` and new password, separated by a colon.<br>For example, `1:MyNewPassword`. |
| **Update password with file** | Update a set of passwords in the `AgentConfig.xml` configuration file, by using an external file. Specify the `Login.Handle` and new password, separated by a colon, one per line.<br>For example, `1:MyNewPassword`. |
| **Local IP** | Use this setting to select the IP address that is displayed for all log sources on systems with multiple network interface cards (NIC). |
| **Originating Computer** | Use this setting to select the IP address that is displayed only for Windows events on systems with multiple NICs. |
| **Version** | Update the `AgentConfig.xml` version number. |

The `installHelper.exe` file has the following update flags:

*Table 18. InstallHelper update flags*

| | |
|---|---|
| `-h [--help]` | Provides detailed information on the `installHelper.exe` usage options. |
| `-P [ --update-password ]` | Update a password in the `AgentConfig.xml` configuration file. Specify the `Login.Handle` and new password, colon separated.<br>For example, `1:MyNewPassword`.<br>**Note:** The password is in plain text. |
| `-F [ --update-password-with-file ]` | Update a set of passwords in the `AgentConfig.xml` configuration file by using an external file. Specify the `Login.Handle` and new password, separated by a colon, one per line.<br>For example, `1:MyNewPassword`.<br>**Note:** Make sure you erase the input file or keep it secured. |
| `-T [ --update-auth-token ]` | The new authentication token to be used to communicate with the configuration server. |
| `-L [localIP]` | Use this setting to select the IP address that is displayed for all log sources on systems with multiple network interface cards (NIC).<br>For example, `installerhelper.exe -L 192.0.2.0` |
| `-O [OrigComputer]` | Use this setting to select the IP address that is displayed for Windows events on systems with multiple NICs.<br>For example, `installerhelper.exe -O 198.51.100.0` |

**Note:** Changing **LocalIP** also changes the **OriginatingComputer** value in the WinCollect syslog header, but does not change the **Log Source Identifier**. If you do not change the value of the **LocalIP**, but add another value in for the **OriginatingComputer**, it overrides the **LocalIP** value after the agent restarts.

For example, to change an authorization token for a WinCollect agent, type the following command in the command prompt of the Windows host:

```
<WinCollect_installation_path>/bin/installHelper.exe -T <authorization_token>
```

**Related concepts**

Local installations with no remote polling
Install WinCollect locally on each host that you cannot remotely poll. After you install WinCollect, IBM Security QRadar automatically discovers the agent and you can create a WinCollect log source.

Windows event subscriptions for WinCollect agents
To provide events to a single WinCollect agent, you can use Windows event subscriptions to forward events. When event subscriptions are configured, numerous Windows hosts can forward their events to IBM Security QRadar without needing administrator credentials.

**Related tasks**

Configuring access to the registry for remote polling

## Local installations with no remote polling

Install WinCollect locally on each host that you cannot remotely poll. After you install WinCollect, IBM Security QRadar automatically discovers the agent and you can create a WinCollect log source.

You can specify to use the local system by selecting the Local System check box in the log source configuration.

Local installations are suitable for domain controllers where the large event per second (EPS) rates can limit the ability to remotely poll for events from these systems. A local installation of a WinCollect agent provides scalability for busy systems that send bursts of events when user activity is at peak levels.

**Related concepts**

Changing WinCollect configuration from the command prompt
You can change the configuration of a WinCollect agent from the command prompt of the Windows host.

Windows event subscriptions for WinCollect agents
To provide events to a single WinCollect agent, you can use Windows event subscriptions to forward events. When event subscriptions are configured, numerous Windows hosts can forward their events to IBM Security QRadar without needing administrator credentials.

**Related tasks**

Configuring access to the registry for remote polling

## Configuring access to the registry for remote polling

Before a WinCollect log source can remotely poll for events, you must configure a local policy for your Windows-based systems.

When a local policy is configured on each remote system, a single WinCollect agent uses the Windows Event Log API to read the remote registry and retrieve event logs. The Windows Event Log API does not require domain administrator credentials. However, the event API method does require an account that has access to the remote registry and to the security event log.

By using this collection method, the log source can remotely read the full event log. However, the method requires WinCollect to parse the retrieved event log information from the remote host against cached message content. WinCollect uses version information from the remote operating system to ensure that the message content is correctly parsed before it forwards the event to IBM Security QRadar.

### Procedure

1. Log on to the Windows computer that you want to remotely poll for events.
2. Select **Start** > **StartPrograms** > **Administrative Tools** and then click **Local Security Policy**.
3. From the navigation menu, select **Local Policies** > **User Rights Assignment**.
4. Right-click **Manage auditing and security log** > **Properties**.
5. From the **Local Security Setting** tab, click **Add User or Group** to add your WinCollect user to the local security policy.
6. Log out of the Windows host and try to poll the remote host for Windows-based events that belong to your WinCollect log source.

   If you cannot collect events for the WinCollect log source, verify that your group policy does not override your local policy. You can also verify that the local firewall settings on the Windows host allow remote event log management.

**Related concepts**

Changing WinCollect configuration from the command prompt
You can change the configuration of a WinCollect agent from the command prompt of the Windows host.

Local installations with no remote polling

Install WinCollect locally on each host that you cannot remotely poll. After you install WinCollect, IBM Security QRadar automatically discovers the agent and you can create a WinCollect log source.

Windows event subscriptions for WinCollect agents
To provide events to a single WinCollect agent, you can use Windows event subscriptions to forward events. When event subscriptions are configured, numerous Windows hosts can forward their events to IBM Security QRadar without needing administrator credentials.

# Windows event subscriptions for WinCollect agents

To provide events to a single WinCollect agent, you can use Windows event subscriptions to forward events. When event subscriptions are configured, numerous Windows hosts can forward their events to IBM Security QRadar without needing administrator credentials.

## Forwarded events

The events that are collected are defined by the configuration of the event subscription on the remote host that sends the events. WinCollect forwards all of the events that are sent by the subscription configuration, regardless of what event log check boxes are selected for the log source.

Windows event subscriptions, or forwarded events, are not considered local or remote, but are event listeners. Use the WinCollect **Forwarded Events** check box to enable the WinCollect log source to identify Windows event subscriptions. Although the WinCollect agent displays only a single log source in the user interface, the log source listens and processes events for potentially hundreds of event subscriptions. One log source in the agent list is for all event subscriptions. The agent recognizes the event from the subscription, processes the content, and then sends the syslog event to QRadar.

**Note:** Forwarded events can be collected with the Forwarded Events check box only. An XPATH cannot be used.

Forwarded events are displayed as `Windows Auth @ <hostname>` or *<FQDN>* in the **Log Activity** tab. Conversely, locally or remotely collected events appear as `Windows Auth @ <IP address>` or *<hostname>*. When WinCollect processes an event that is collected locally or remotely, it includes an extra syslog header that identifies the event as a WinCollect event. Because the forwarded event is a pass-through or listener, forwarded events don't include the WinCollect identifier and appear as standard events.

**Important:** WinCollect collects only those forwarded events that appear in the Windows Event Viewer.

## Supported software environments

Event subscriptions apply only to WinCollect agents and hosts that are configured on the following Windows operating systems:

- Windows 8 (most recent)
- Windows Server 2012 (most recent)
- Windows 10 (most recent)
- Windows Server 2016 (including Core)
- Windows Server 2019 (including Core)

**Important:** WinCollect is not supported on versions of Windows that are designated end-of-life by Microsoft. After software is beyond the Extended Support End Date, the product might still function as expected. However, IBM does not make code or vulnerability fixes to resolve WinCollect issues for older operating systems. For example, Microsoft Windows Server 2003 R2 and Microsoft Windows XP are operating systems that are beyond the "Extended Support End Date." Any questions about this announcement can be discussed in the IBM QRadar Collecting Windows Events (`WMI/ALE/WinCollect`) forum. For more information, see https://support.microsoft.com/en-us/lifecycle/search (https://support.microsoft.com/en-us/lifecycle/search).

For more information about event subscriptions, see your Microsoft documentation or the Microsoft technical website (http://technet.microsoft.com/en-us/library/cc749183.aspx).

## Troubleshooting event collection

Microsoft event subscriptions don't have an alert mechanism to indicate when an event source stopped sending events. If a subscription fails between the two Windows systems, the subscription appears active, but the service that is responsible for the subscription can be in an error state. With WinCollect, the remotely polled or local log sources can time out when events are not received within 720 minutes (12 hours).

**Related concepts**

Changing WinCollect configuration from the command prompt
You can change the configuration of a WinCollect agent from the command prompt of the Windows host.

Local installations with no remote polling
Install WinCollect locally on each host that you cannot remotely poll. After you install WinCollect, IBM Security QRadar automatically discovers the agent and you can create a WinCollect log source.

**Related tasks**

Configuring access to the registry for remote polling

# Configuring Microsoft event subscriptions

Configure Microsoft event subscriptions to forward events to a single WinCollect agent.

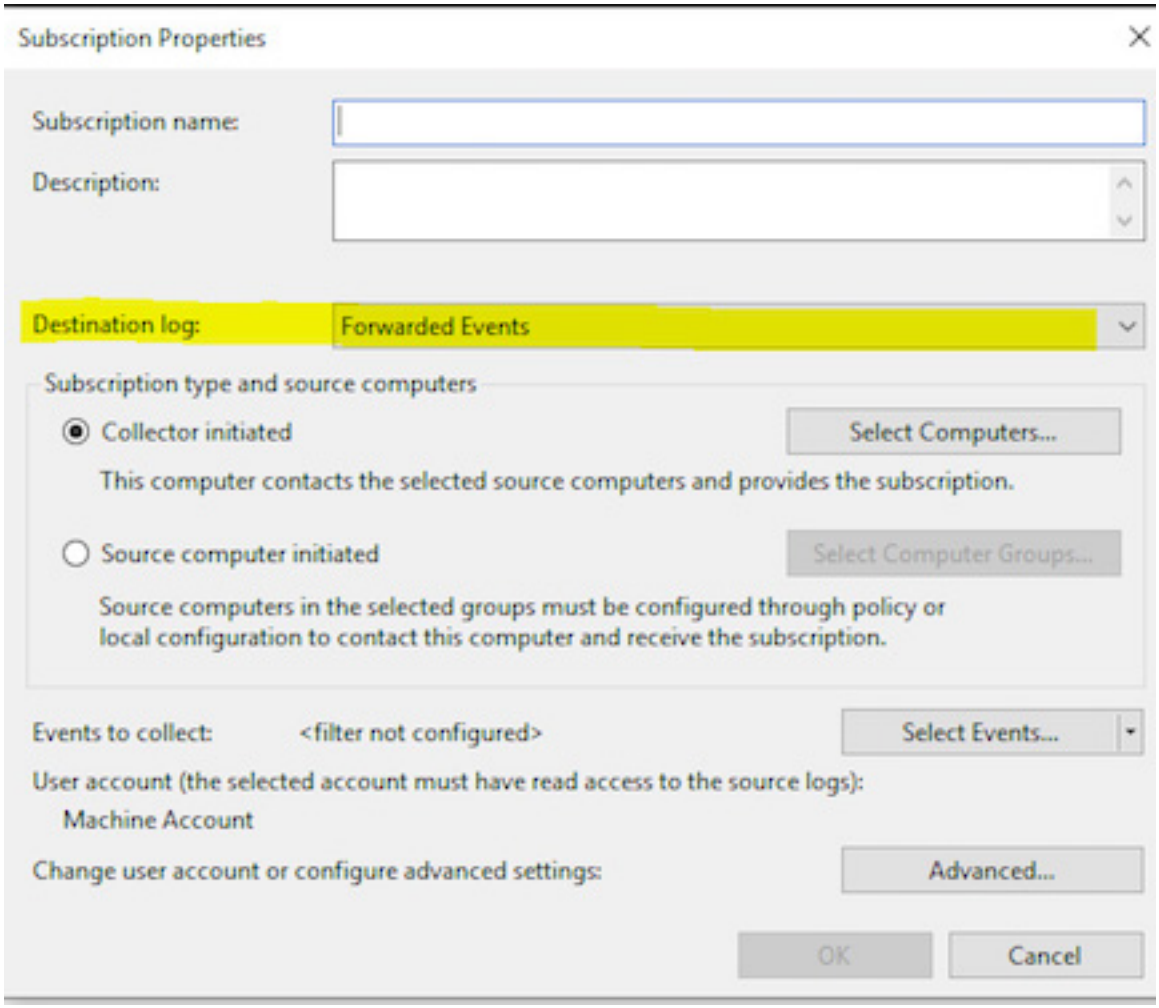
## Before you begin

WinCollect supports event subscriptions with the following parameters:

**Forwarded Events**
The subscription must send the logs to the forwarded event channel. Selected in the **Destination log** list (see screen capture).

**Subscriptions**
The subscription configured to use **ContentFormat:** `RenderedText` and **Locale:** `en-US`

**Locale**
Locale must be `en_US` for the Windows computer where WinCollect is installed.




**Note:** If you are using domain controllers, consider installing local WinCollect agents on the servers. Due to the potential number of generated events, use a local log source with the agent that is installed on the domain controller.

## Procedure

1. Configure event subscriptions on your Windows hosts.

   For instructions on configuring event subscriptions, see the Microsoft Event Collector documentation. (https://docs.microsoft.com/en-us/windows/desktop/wec/creating-an-event-collector-subscription)
2. Configure a log source on the WinCollect agent that receives the events.

   You must select the **Local System** check box and **Forwarded Events** check box for the WinCollect log source.

   **Note:** IBM Support does not support the creation or maintenance of Microsoft Subscriptions.

**Related tasks**

Adding a log source to a WinCollect agent

When you add a new log source to a WinCollect agent or edit the parameters of a log source, the WinCollect service is restarted. The events are cached while the WinCollect service restarts on the agent.

# Chapter 5. Log sources for WinCollect agents

A WinCollect agent can collect and forward events from the local system, or remotely poll a number of Windows-based log sources and operating systems for their events.

You can add log sources that communicate through a WinCollect agent individually for remote polling. If the log sources contain similar configurations, you can simultaneously add multiple, or bulk add log sources. A change to an individually added log source updates only the individual log source. A change that you make to a group of log sources updates all of the log sources in the log source group.

You can add a local log source for local collection. You can create a log source manually, if it wasn't autocreated.

**Important:** If your deployment has the same user name accounts on different domains, ensure that you configure domain information when you create the WinCollect log source.

## Windows event logs

You can collect the event logs from your Windows endpoints.

When you query a Windows event log, the query includes every event in the log. You can use event log filtering or XPath queries to limit the events that you receive.

Windows event logs are supported in the following languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Russian
- Spanish

### Windows event log filtering

You can configure the WinCollect agent to ignore or to include specific events collected from the Windows event log. You can limit the total EPS (events per second) that are sent to the QRadar Console by using the filter types.

The WinCollect agents can be configured to ignore events globally by ID code or log source. Global exclusions use the **EventIDCode** field from the event payload. To determine the values that are excluded, source and ID exclusions use the **`Source=field`** and the **`EventIDCode=field`** of the Windows payload. Separate multiple sources by using a semi-colon. Events filters such as exclusion, inclusion, and NSA are available for the following log source types:

- Security
- System
- Application
- DNS Server
- File Replication Service

- Directory Service
- Forwarded Events

The WinCollect agent requests all available events from the Event Collection API each time the value specified in the Polling Interval field expires.

For the exclusion filter, the agent examines all of the events retrieved from the Event Collection API and ignores events that match the exclusions defined by the administrator (either by Windows Event ID or by source). The agent then takes the remaining events and assembles the **`name=value`** pairs and forwards the events to either the QRadar Console or the Event Collector appliance. However, for the inclusion filter, the agents pulls events that matches the Event IDs specified by the administrator and forward those events to QRadar Console or Event Collector.

The NSA filter is a unique type of filter that includes a corresponding list of pre-defined security Event IDs, which the agent pulls from the Security, System, Application and DNS logs. These pre-defined security Event IDs are included in the events that the agent forwards to the Console or Event Collector.

**Tip:** The Forwarded Events filter requires you to identify the source or channel, with the eventIDs that you wish to filter in parentheses. Use semicolons as delimiters. For example:

```
Application(200-256,4097,34);Security(1);Symantec(1,13)
```

In this example, event IDs from 200 to 256, 4097 and 34 are filtered for the channel Application, event ID 1 is filtered for Security, and event IDs 1 and 13 are filtered for the source called Symantec.

## Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

*Table 19. Common WinCollect log source parameters*

| Parameter | Description |
|---|---|
| **Log Source Identifier** | The IP address or hostname of a remote Windows operating system from which you want to collect Windows-based events. The log source identifier must be unique for the log source type.<br>Used to poll events from remote sources. |
| **Local System** | Disables remote collection of events for the log source.<br>The log source uses local system credentials to collect and forward events to QRadar.<br>**Note:** You must clear this box if you are using a fully qualified domain name (FQDN) log source identifier and the agent is installed on a domain controller. |
| **Domain** | Optional<br>The domain that includes the Windows-based log source.<br>The following examples use the correct syntax: `LAB1`, `server1.mydomain.com` The following syntax is incorrect: `\\mydomain.com` |

*Table 19. Common WinCollect log source parameters (continued)*

| Parameter | Description |
|---|---|
| **Event Rate Tuning Profile** | For the default polling interval of 3000 ms, the approximate Events per second (EPS) rates attainable are as follows:<br>• **Default (Endpoint)**: 33-50 EPS<br>• **Typical Server**: 166-250 EPS<br>• **High Event Rate Server**: 416-625 EPS<br>For a polling interval of 1000 ms the approximate EPS rates are as follows:<br>• **Default (Endpoint)**: 100-150 EPS<br>• **Typical Server**: 500-750 EPS<br>• **High Event Rate Server**: 1250-1875 EPS<br>For more information about tuning WinCollect, see IBM Support (http://www.ibm.com/support/docview.wss?uid=swg21672193). |
| **Polling Interval (ms)** | The interval, in milliseconds, between times when WinCollect polls for new events. |
| **Application or Service Log Type** | Optional.<br>Used for XPath queries.<br>Provides a specialized XPath query for products that write their events as part of the Windows application log. Therefore, you can separate Windows events from events that are classified to a log source for another product. |
| **Event Log Poll Protocol** | The protocol that QRadar uses to communicate with the Windows device. The default is **MSEVEN6**. |

*Table 19. Common WinCollect log source parameters (continued)*

| Parameter | Description |
|---|---|
| **Log Filter Type** | Configures the WinCollect agent to ignore specific events from the Windows event log.<br><br>You can also configure WinCollect agents to ignore events globally by ID code or log source.<br><br>*Exclusion filters* for events are available for the following log source types: Security, System, Application, DNS Server, File Replication Service, and Directory Service<br><br>Global exclusions use the **EventIDCode** field from the event payload. To determine the values that are excluded, source and ID exclusions use the `Source=` field and the `EventIDCode=` field of the Windows event payload. Separate multiple sources by using a semi-colon.<br><br>**Example:** Exclusion filters can use commas and hyphens to filter single EventIDs or ranges, such as 4609, 4616, 6400-6405.<br><br>For more information about filtering, see WinCollect Event Filtering (http://www.ibm.com/support/docview.wss?uid=swg21672656). |
| **Security** | Select the checkbox to enable WinCollect to forward security logs to QRadar. |
| **Security Log Filter Type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br><br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br><br>The **NSA Filter** option populates the **Security Log Filter** field with a list of event IDs recommended by the National Security Agency.<br><br>The default is **No Filtering**.<br><br>**Note:** If you select a filter type from the list, a new field **Security Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **System** | Select the checkbox to enable WinCollect to forward system logs to QRadar. |

*Table 19. Common WinCollect log source parameters (continued)*

| Parameter | Description |
|---|---|
| **System Log Filter Type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br>The **NSA Filter** option populates the **System Log Filter** field with a list of event IDs recommended by the National Security Agency.<br>The default is **No Filtering**.<br>**Note:** If you select a filter type from the list, a new field **System Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **Application** | Select the checkbox to enable WinCollect to forward application logs to QRadar. |
| **Application Log Filter Type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br>The **NSA Filter** option populates the **Application Log Filter** field with a list of event IDs recommended by the National Security Agency.<br>The default is **No Filtering**.<br>**Note:** If you select a filter type from the list, a new field **Application Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **DNS Server** | Select the checkbox to enable WinCollect to forward DNS Server logs to QRadar. |
| **DNS Server Log Filter Type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br>The **NSA Filter** option populates the **DNS Server Log Filter** field with a list of event IDs recommended by the National Security Agency.<br>The default is **No Filtering**.<br>**Note:** If you select a filter type from the list, a new field **DNS Server Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **File Replication Service** | Select the checkbox to enable WinCollect to forward File Replication Service logs to QRadar. |

| *Table 19. Common WinCollect log source parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **File Replication Service Log Filter Type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br>**Note:** If you select a filter type from the list, a new field **File Replication Service Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **Directory Service** | Select the checkbox to enable WinCollect to forward Directory Service logs to QRadar. |
| **Directory Service Log Filter Type** | To ignore specific events ID collected from the Windows event log, select the **Exclusion Filter**.<br>To include specific events ID collected in the Windows event log, select the **Inclusion Filter**.<br>**Note:** If you select a filter type from the list, a new field **Directory Service Log Filter** displays. You must provide the event IDs that you want to include or exclude. |
| **Forwarded Events** | Enables QRadar to collect events that are forwarded from remote Windows event sources that use subscriptions.<br>Forward events that use event subscriptions are automatically discovered by the WinCollect agent and forwarded as if they are a syslog event source.<br>When you configure event forwarding from your Windows system, enable event pre-rendering.<br>**Important:** WinCollect supports pulling logs only from the Forwarded Events channel. Writing events from a subscription to a different channel is not supported. |

*Table 19. Common WinCollect log source parameters (continued)*

| Parameter | Description |
|---|---|
| **Forwarded Events filter type** | To ignore specific events ID collected from the Windows event log, select **Exclusion Filter**.<br><br>To include specific events ID collected in the Windows event log, select **Inclusion Filter**.<br><br>The **NSA Filter** option populates the **Forwarded Events filter** field with all channels and their respective filters, as recommended by the National Security Agency.<br><br>The default is **No Filtering**.<br><br>**Note:** If you select a filter type from the list, a new field **Forwarded Events Filter** displays. You must provide the event IDs that you want to include or exclude.<br><br>The Forwarded Events filter requires you to identify the source or channel, with the eventIDs that you want to filter in parentheses. Use semicolons as delimiters. For example:<br><br>`Application(200-256,4097,34);`<br>`Security(1);Symantec(1,13)`<br><br>In this example, event IDs 200 - 256, 4097 and 34 are filtered for the channel Application. Event ID 1 is filtered for Security. Event IDs 1 and 13 are filtered for the source called Symantec. |
| **Event Types** | At least one event type must be selected.<br><br>If you need to collect specific event types, follow the instructions for creating a custom XPath with those specific event types. For more information, see “Creating a custom view” on page 64. |
| **Enable Active Directory Lookups** | If the WinCollect agent is in the same domain as the domain controller that is responsible for the Active Directory lookup, you can select this checkbox. If you do, leave the override domain and DNS parameters blank.<br><br>**Important:** You must enter values for the **Domain Controller Name Lookup** and **DNS Domain Name Lookup** parameters. |
| **Override Domain Controller Name** | Required when the domain controller that is responsible for Active Directory lookup is outside of the domain of the WinCollect agent.<br><br>The IP address or hostname of the domain controller that is responsible for the Active Directory lookup. |

| *Table 19. Common WinCollect log source parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **XPath Query** | Structured XML expressions that you use to retrieve customized events from Windows event logs.<br>If you specify an XPath query to filter events, the check boxes that you selected from the **Standard Log Type** or **Event Type** are collected along with the XPath Query.<br>To collect information by using an XPath Query, you might be required to enable **Remote Event Log Management** on Windows 2008. |
| **Target Internal Destination** | Use any managed hosts with an event processor component as an internal destination. |
| **Target External Destination** | Forwards your events to one or more external destinations that you configured in your destination list. |

# Applications and Services logs

Use XPath queries to collect events from the Applications and Services event logs.

XPath queries are structured XML expressions that you use to retrieve customized events from the Windows event logs.

**Related reference**

Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

## Creating a custom view

Use the Microsoft Event Viewer to create custom views, which can filter events for severity, source, category, keywords, or specific users.

WinCollect log sources can use XPath filters to capture specific events from your logs. To create the XML markup for your XPath Query parameter, you must create a custom view. You must log in as an administrator to use Microsoft Event Viewer.

**Note:** Using more than 10 XPath queries can affect WinCollect performance, depending on the XPath and the number of events coming in to each channel.

XPath queries that use the WinCollect protocol the TimeCreated notation do not support filtering of events by a time range. Filtering events by a time range can lead to errors in collecting events.

### Procedure

1. On your desktop, select **Start** > **Run**.
2. Type the following command:

   ```
   Eventvwr.msc
   ```
3. Click **OK**.
4. If you are prompted, type the administrator password and press Enter.
5. Click **Action** > **Create Custom View**.

When you create a custom view, do not select a time range from the **Logged** list. The **Logged** list includes the **TimeCreated** element, which is not supported in XPath queries for the WinCollect protocol.

6. In **Event Level**, select the check boxes for the severity of events that you want to include in your custom view.
7. Select an event log source. You can select the source from the **Event sources** drop-down menu, or you can browse to a source from the **Event logs** drop-down menu.
8. Type the event IDs to filter from the event or log source.

   Use commas to separate IDs.

   The following list contains an individual ID and a range: 4133, 4511-4522
9. From the **Task Category** list, select the categories to filter from the event or log source.
10. From the **Keywords** list, select the keywords to filter from the event or log source.
11. Type the user name to filter from the event or log source.
12. Type the computer or computers to filter from the event or log source.
13. Click the **XML** tab.
14. Copy and paste the XML to the **XPath Query** field of your WinCollect log source configuration

### What to do next

Configure a log source with the XPath query. For more information, see "Applications and Services logs" on page 64.

## XPath query examples

Use XPath examples for monitoring events and retrieving logon credentials, as a reference when you create XPath queries.

For more information about XPath queries, see your Microsoft documentation.

**Note:** XPath uses only the MSEVEN6 event protocol.

### Example: Monitoring events for a specific user

In this example, the query retrieves events from all Windows event logs for the guest user.

**Important:** XPath queries cannot filter Windows Forwarded Events.

```
<QueryList>
<Query Id="0" Path="Application">
<Select Path="Application">*[System[(Level=4 or Level=0) and
Security[@UserID='S-1-5-21-3709697454-1862423022-1906558702-501
']]]</Select>
<Select Path="Security">*[System[(Level=4 or Level=0) and
Security[@UserID='S-1-5-21-3709697454-1862423022-1906558702-501
']]]</Select>
<Select Path="Setup">*[System[(Level=4 or Level=0) and
Security[@UserID='S-1-5-21-3709697454-1862423022-1906558702-501
']]]</Select>
<Select Path="System">*[System[(Level=4 or Level=0) and
Security[@UserID='S-1-5-21-3709697454-1862423022-1906558702-501
']]]</Select>

</Query>
</QueryList>.
```

### Example: Credential logon for Windows 2008

In this example, the query retrieves specific event IDs from the security log for Information-level events that are associated with the account authentication in Windows 2008.

```
<QueryList>
<Query Id="0" Path="Security">
```

```
<Select Path="Security">*[System[(Level=4 or Level=0) and
( (EventID &gt;= 4776 and EventID &lt;= 4777) )]]</Select>
</Query>
</QueryList>
```

*Table 20. Event IDs used in credential logon example*

| ID | Description |
|---|---|
| 4776 | The domain controller attempted to validate credentials for an account. |
| 4777 | The domain controller failed to validate credentials for an account. |

## Example: Retrieving events based on user

In this example, the query examines event IDs to retrieve specific events for a user account that is created on a fictional computer that contains a user password database.

```
<QueryList>
<Query Id="0" Path="Security">
<Select Path="Security">*[System[(Computer='Password_DB') and
(Level=4 or Level=0) and (EventID=4720 or (EventID &gt;= 4722
and EventID &lt;= 4726) or (EventID &gt;= 4741 and EventID
&lt;= 4743) )]]</Select>
</Query>
</QueryList>
```

*Table 21. Event IDs used in database example*

| ID | Description |
|---|---|
| 4720 | A user account was created. |
| 4722 | A user account was enabled. |
| 4723 | An attempt was made to change the password of an account. |
| 4724 | An attempt was made to reset password of an account. |
| 4725 | A user account was disabled. |
| 4726 | A user account was deleted. |
| 4741 | A user account was created. |
| 4742 | A user account was changed. |
| 4743 | A user account was deleted. |

## Example: Retrieving DNS analytic logs

In this example, the query retrieves all events that are captured in DNS analytic logs.

```
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-DNSServer/Analytical">
    <Select Path="Microsoft-Windows-DNSServer/Analytical">*</Select>
  </Query>
</QueryList>
```

### Example: Retrieving events with Sysinternals Sysmon

In this example, the query retrieves all events that are captured by SysInternals Sysmon.

```
<QueryList>
<Query Id="0" Path="Microsoft-Windows-Sysmon/Operational">
<Select Path="Microsoft-Windows-Sysmon/Operational">*</Select>
</Query>
</QueryList>
```

## Microsoft DHCP log source configuration options

Use this reference information to configure the WinCollect plug-in for Microsoft DHCP.

**Restriction:** The WinCollect agent must be in the same time zone as the remote DHCP server that it is configured to poll.

*Table 22. Microsoft DHCP protocol parameters*

| Parameter | Description |
|---|---|
| **Log Source Type** | **Microsoft DHCP** |
| **Protocol Configuration** | **WinCollect Microsoft DHCP** |
| **Local System** | The WinCollect agent must be installed on the Microsoft DHCP Server.<br>The log source uses local system credentials to collect and forward events to QRadar. |

For more information about DHCP log source configuration, see the *IBM QRadar DSM Configuration Guide*.

*Table 23. Default root log directory paths for Microsoft DHCP events.*

The DHCP event logs that are monitored by WinCollect are defined by the directory path that you specify in your WinCollect DHCP log source.

| Collection type | Root log directory |
|---|---|
| Local | `c:\WINDOWS\system32\dhcp` |
| Remote | `\\DHCP IP address\c$ \Windows\System32\dhcp` |

*Table 24. Example log format for Microsoft DHCP events.*

WinCollect evaluates the root log directory folder to automatically collect new DHCP events that are written to the event log. DHCP event logs start with DHCP, contain a three-character day of the week abbreviation, and end with a `.log` file extension. Any DHCP log files that are in the root log directory and match either an IPv4 or IPv6 DHCP log format are monitored for new events by the WinCollect agent.

| Log type | Example of log file format |
|---|---|
| IPv4 | `DhcpSrvLog-Mon.log` |
| IPv6 | `DhcpV6SrvLog-Wed.log` |

**Related reference**

Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

# Microsoft Exchange Server log source configuration options

Use this reference information to configure the WinCollect plug-in for Microsoft Exchange Server.

## Supported versions

WinCollect supports the following versions of Microsoft Exchange :

- Microsoft Exchange 2003
- Microsoft Exchange 2007
- Microsoft Exchange 2010
- Microsoft Exchange 2013
- Microsoft Exchange 2016
- Microsoft Exchange 2019

*Table 25. Microsoft Exchange Server protocol parameters*

| Parameter | Description |
|---|---|
| **Log Source Type** | **Microsoft Exchange Server** |
| **Protocol Configuration** | **WinCollect Microsoft Exchange** |
| **Local System** | The WinCollect agent must be installed on the Microsoft Exchange Server.<br>The log source uses local system credentials to collect and forward events to QRadar. |

Ensure that the firewalls that are located between the Exchange Server and the remote host allow traffic on the following ports:

- TCP port 135 for Microsoft Endpoint Mapper.
- UDP port 137 for NetBIOS name service.
- UDP port 138 for NetBIOS datagram service.
- TCP port 139 for NetBIOS session service.
- TCP port 445 for Microsoft Directory Services to transfer files across a Windows share.

For more information about Microsoft Exchange log source configuration, see The *IBM QRadar DSM Configuration Guide*.

*Table 26. Default OWA directory paths for Microsoft Exchange Server events.*

The Exchange Server OWA event logs that are monitored by WinCollect are defined by the directory path that you specify in your WinCollect Exchange Server log source. Microsoft Exchange writes to two directories: `W3SVC1` and `W3SVC2`. The Microsoft Exchange plug-in monitors all recursive files under the `C:\inetpub\logs\LogFiles\` directory.

| Collection type | Root log directory |
|---|---|
| Local | `C:\inetpub\logs\LogFiles\W3SVC1` |
| Remote | `\\<Exchange Server IP address>\C$ \inetpub\logs\LogFiles\W3SVC1` |

*Table 27. Default Message Tracking directory paths for Microsoft Exchange Server events.*

The Exchange Server Message Tracking event logs that are monitored by WinCollect are defined by the directory path that you specify in your WinCollect Exchange Server log source.

| Collection type | Root log directory |
|---|---|
| Local | `C:\Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\Message Tracking` |
| Remote | `\\<Exchange Server IP address>\C$ \Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\Message Tracking` |

*Table 28. Default SMTP/Mail directory paths for Microsoft Exchange Server events.*

The Exchange Server SMTP/Mail event logs that are monitored by WinCollect are defined by the directory path that you specify in your WinCollect Exchange Server log source.

| Collection type | Root log directory |
|---|---|
| Local | `C:\Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\Hub\ProtocolLog` |
| Remote | `\\<Exchange Server IP address>\C$ \Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\Hub\ProtocolLog` |

## DNS debug log source configuration options

Use the reference information to configure the WinCollect plug-in for Microsoft Windows DNS debug logging.

**Important:** DNS debug logging can affect system performance and disk space because it provides detailed data about information that the DNS server sends and receives. Enable DNS debug logging only when you require this information.

DNS debug logging is supported on the following Windows versions:

- Windows Server 2019 (including Core)
- Windows Server 2016 (including Core)
- Windows Server 2012 R2
- Windows Server 2012

| *Table 29. DNS debug protocol parameters* | |
|---|---|
| **Parameter** | **Description** |
| **File Reader Type** | Reads file contents. Both options have basic Unicode encoding support for byte-order marks.<br>If you choose the **Text (file held open)** option, then WinCollect maintains a shared read and write lock on the monitored log file.<br>If you choose the **Text (file open when reading)** option, then WinCollect maintains a shared read and write lock on the log file only when it reads the file. |
| **File Monitor Type** | Detects file and directory changes:<br>The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your DNS log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote DNS log and compares the file to the last polling interval. If the log contains new entries, the entries are retrieved. |
| **File Pattern** | The regular expression (regex) required to match the DNS debug log file set in the DNS manager. |
| **Root Directory** | The directory where WinCollect monitors files. The directory must be Local File System for local collection, or a valid Microsoft Windows universal naming convention (UNC) path for remote collection.<br>This value must match the file path that is configured in your DNS manager.<br>**Important:** Due to restrictions in distributed systems, the path can't be verified in the user interface. |
| **Include DNS Details** | Includes DNS details in the Windows Server DNS debugging log. |

For more information about Microsoft DNS Debug specifications, see the *IBM QRadar DSM Configuration Guide*.

## Enabling DNS debugging on Windows Server

Enable DNS debugging on Windows Server to collect information that the DNS server sends and receives.

### Before you begin

The DNS role must be installed on the Windows Server.

### Procedure

1. Open the DNS Manager with the following command:

```
dnsmgmt.msc
```

2. Right-click the DNS server and click **Properties**.
3. Click the **Debug Logging** tab.
4. Select **Log packets for debugging**.
5. In the log file, type the file path and name, and the maximum size.

   **Important:** The file path and name, must align with the **Root Directory** and **File Pattern** that you provided when you configured the Microsoft DNS log source.
6. If you want to include DNS details in the log, select **Details** in the **Other options** section, and then select **Include DNS Details** in the log source.
7. Click **Apply** and then click **OK**.

# Collecting DNS Analytic Logs by using XPath

To collect DNS Analytic logs by using WinCollect, you must first configure Windows to collect analytic logs and then add an XPath to the WinCollect Agent log source to collect the logs and send them to QRadar.

## About this task

Use Event Viewer to configure Windows to collect DNS Server analytic logs.

## Procedure

1. To open the Event Viewer, type `eventvwr.msc` at an elevated command prompt, and press **Enter**.
2. Go to `Applications and Services Logs\Microsoft\Windows\DNS-Server`.
3. Right-click **DNS-Server**, and then click **View** > **Show Analytic and Debug Logs**.
4. Right-click the **Analytical** log, and then click **Properties**.
5. In the **When maximum event log size is reached** section, choose **Do not overwrite events (Clear logs manually)**, select **Enable logging**, and then click **OK** on the resulting dialog box.

   **Important:** If you do not select this option, the WinCollect Agent can't collect the Analytical log, because the logs are stored in etl format. For more information, see https://support.microsoft.com/en-ca/help/2488055/error-when-enabling-analytic-or-debug-event-log.
6. Click **OK** to enable the DNS Server Analytic event log.

   **Attention:** You must manually clear the logs and restart the agent when the event log is full.
7. In the log source, add the following XPath to the WinCollect Agent:

```
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-DNSServer/Analytical">
    <Select Path="Microsoft-Windows-DNSServer/Analytical">*</Select>
  </Query>
</QueryList>
```

# File Forwarder log source configuration options

Use the reference information to configure the WinCollect plug-in for the File Forwarder log source.

You must also configure parameters that are not specific to this plug-in. The File Forwarder plug-in can be used with Universal DSM to poll many types of logs from the Windows host.

*Table 30. File Forwarder protocol parameters*

| Parameter | Description |
|---|---|
| **Log Source Type** | **Universal DSM** |

| *Table 30. File Forwarder protocol parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Protocol Configuration** | Select **WinCollect File Forwarder**. |
| **Local System** | Disables remote collection of events for the log source. The log source uses local system credentials to collect and forward events to QRadar. |
| **Root Directory** | The location of the log files to forward to QRadar.<br>If the WinCollect agent remotely polls for the file, the root log directory must specify both the server and the folder location for the log files.<br>**Example:** \ \server\sharedfolder\remotelogs\ |
| **Filename Pattern** | The regular expression (regex) that is required to filter the file names. All files that match the pattern are included in the processing. The default file pattern is .* and matches all files in the Root Directory. |
| **Monitoring Algorithm** | The **Continuous Monitoring** option is intended for files systems that append data to log files.<br>The **File Drop** option is used for the log files in the root log directory that are read one time, and then ignored in the future. |
| **Only Monitor Files Created Today** | Enabled by default. Clear this option to monitor files from before the current day. |
| **File Monitor Type** | The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your event log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote event log and compares the file to the last polling interval. If the event log contains new events, the event log is retrieved. |

| *Table 30. File Forwarder protocol parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **File Reader Type** | If you choose the **Text (file held open)** option, the system that generates your event log continually leaves the file open to append events to the end of the file.<br>If you choose the **Text (file open when reading)** option, the system that generates your event log opens the event log from the last known position, and then writes events and closes the event log.<br>Select the **Memory Mapped Text (local only)** option only when advised by IBM Professional Services. This option is used when the system that generates your event log polls the end of the event log for changes. This option requires that you also select the **Local System** check box. |
| **File Reader Encoding** | For files without a BOM, select **ANSI** if you want the files converted to UTF8. Otherwise, select **UTF8** if the files are already in UTF8 and no conversion is needed. |
| **File Parser Type** | Files can be parsed in two ways: Single Line or Multi Line.<br>**Single Line**<br>Parses a file and creates an event for each line.<br>**Multi Line**<br>Parses an XML file and creates an event that comprises multiple lines from the point that a specified starting token is parsed, until the next time the specified starting token is parsed.<br>**Note:** Multi Line parsing currently only supports XML file types. |
| **Multi Line "Starts With" Regex Token** | The Multi Line File Parser Type requires a "Starts With" token. The "Starts With" token should be the regex that is required to identify every character from the beginning of the line you want to start a multi line event with. It is important to make your regex as accurate as possible to avoid combining events due to similar whitespace before the characters, and to avoid not parsing the file at all due to not finding a "Starts With" token. |

Example for Multi Line parser type XML file

To ensure that the XML file is parsed to generate an event for every `<event>` node, use a multi-line "Starts With token of "`\s*<event>`."

```
<EventList>
    <event>
        <timeStamp=10101010101 payload=example1>
    </event>
    <event>
        <timeStamp=10101010102 payload=example2>
    </event>
```

```
    <event>
        <timeStamp=10101010103 payload=example3>
    </event>
    <event>
        <timeStamp=10101010104 payload=example4>
    </event>
</EventList>
```

The multi-line file parser produces 4 individual events, instead of producing 14 individual single line events. The payload message for the first event created would look like this example:

```
<event> <timeStamp=10101010101 payload=example1> </event>
```

**Note:** A multi-line 'Starts With' token " `<event>`" would also work; however tabs and spaces can look the same and be coded differently. Using "`\s*<event>`" is a better option, because it covers both types of white space.

**Related reference**

Windows log source parameters
Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

# Microsoft IAS log source configuration options

Use the reference information to configure the WinCollect plug-in for Microsoft IAS.

*Table 31. Supported Windows versions and log formats*

| Microsoft IAS | Supported Versions |
|---|---|
| Microsoft Windows support | Windows Server 2019<br>Windows Server 2016<br>Windows Server 2012 R2 |
| NPS® log server log formats | Data Transformation Service<br>Open Database Connectivity<br>Internet Authentication Service |

**Important:** WinCollect does not support events that are logged to a Microsoft SQL Server.

## Microsoft IAS directory structure for event collection

The event logs that are monitored by WinCollect are defined by the root directory that you should configure in your log source.

When you specify a root log directory, you must point the WinCollect agent to the folder that contains your Microsoft IAS or NPS events. The root log directory does not recursively search sub-directories for event files.

To improve performance, you can create a sub folder for your IAS and NPS event logs, for example, `\WINDOWS\System32\Logfiles\NPS`. When you create a specific event folder, the agent does not have to evaluate many files to locate your event logs.

If your system generates a large number of IAS or NPS events, you can configure your Windows system to create a new event log at daily intervals. This action ensures that agents do not have to search large logs for new events.

| *Table 32. Event log default directory structure for Microsoft IAS* | |
|---|---|
| **Event version** | **Root Log Directory** |
| Microsoft Windows Server 2019 | `\Windows\System32\Logfiles\` |
| Microsoft Windows Server 2016 | `\Windows\System32\Logfiles\` |
| Microsoft Windows Server 2012 R2 | `\Windows\System32\Logfiles\` |

### Microsoft IAS protocol parameters

| *Table 33. Microsoft IAS parameters* | |
|---|---|
| **Parameter** | **Description** |
| **Log Source Type** | **Microsoft IAS Server** |
| **Protocol Configuration** | **WinCollect Microsoft IAS / NPS** |
| **Local System** | To collect local events, the WinCollect agent must be installed on the same host as your Microsoft DHCP Server.<br>The log source uses local system credentials to collect and forward events to QRadar. |
| **File Monitor Policy** | The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your event log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote event log and compares the file to the last polling interval. If the event log contains new events, the event log is retrieved. |
| **Polling Interval** | The amount of time between queries to the root log directory for new events. |

**Related reference**

Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

## WinCollect Microsoft IIS log source configuration options

You can configure a log source to use the Microsoft Internet Information Services (IIS). This WinCollect plugin supports a single point of collection for W3C format log files that are on a Microsoft IIS web server.

### Overview for the WinCollect plug-in for Microsoft IIS

You can use one of two methods to collect Microsoft IIS logs with WinCollect. You can install an agent locally on your Microsoft IIS server and configure it accordingly. Or, with WinCollect 7.2.8 and later, you can configure a WinCollect agent to remotely poll the IIS logs. See Table 1 for setting up the directory paths based off your method of log collection.

The WinCollect plug-in for Microsoft IIS can read and forward events for the following logs:

- Website (W3C) logs
- File Transfer Protocol (FTP) logs

- Simple Mail Transfer Protocol (SMTP) logs
- Network News Transfer Protocol (NNTP) logs

The WinCollect plug-in for Microsoft IIS can monitor W3C, IIS, and NCSA formatted event logs. However, the IIS and NCSA event formats do not contain as much event information in their event payloads as the W3C event format. To collect the maximum information available, configure your Microsoft IIS Server to write events in W3C format. WinCollect can collect both ASCII and UTF-8 encoded event log files.

## Supported versions of Microsoft IIS

The Microsoft IIS plug-in for WinCollect supports the following Microsoft IIS software versions:

- Microsoft IIS Server 7.0
- Microsoft IIS Server 7.5
- Microsoft IIS Server 8.0
- Microsoft IIS Server 8.5
- Microsoft IIS Server 10

## WinCollect Microsoft IIS parameters

*Table 34. Microsoft IIS parameters*

| Parameter | Description |
|---|---|
| Protocol Configuration | Select **WinCollect Microsoft IIS**. |
| Log Source Identifier | The IP address or host name of your Microsoft IIS server.<br>It must be unique for the log source type. |
| Root Directory | The directory path to your Microsoft IIS log files.<br>For Microsoft 7.0-10.0 (full site), use:<br>• Local: `%SystemDrive%\inetpub\logs\LogFiles`<br>• Remote: `\\HostnameorIP\c$\inetpub\logs\LogFiles`<br>For Microsoft IIS 7.0-10.0 (individual site), use:<br>• Local: `%SystemDrive%\inetpub\logs\LogFiles\site name`<br>• Remote: `\\HostnameorIP\c$ \inetpub\logs\LogFiles\site name` |
| Polling Interval | The amount of time between queries to the root log directory for new events.<br>The default polling interval is 5000 milliseconds. |
| FTP | Collects File Transfer Protocol (FTP) events from Microsoft IIS. |
| NNTP/News | Collects Network News Transfer Protocol (NNTP) events from Microsoft IIS. |
| SMTP/Mail | Collects Simple Mail Transfer Protocol (SMTP) events from Microsoft IIS. |
| W3C | Collects website (W3C) events from Microsoft IIS. |
| WinCollect Agent | Manages the WinCollect agent log source. |

For more information about configuring a Microsoft IIS log source, see the *IBM QRadar DSM Configuration Guide*.

# Microsoft ISA log configuration options

Use the reference information to configure the WinCollect plug-in for Microsoft ISA.

## Supported versions of Microsoft ISA

The Microsoft ISA plug-in for WinCollect supports the following software versions:

- Microsoft ISA Server 2006
- Microsoft Forefront Threat Management Gateway 2010

## Supported Microsoft ISA or TMG server log formats

Microsoft ISA and Forefront Threat Management Gateway installations create individual firewall and web proxy event logs in a common log directory. To collect these events with WinCollect, you must configure your Microsoft ISA or Microsoft Time Management Gateway to write event logs to a log directory.

**Restriction:** Events that log to a Microsoft SQL server database are not supported by WinCollect.

WinCollect supports the following event log formats:

- Web proxy logs in WC3 format (w3c_web)
- Microsoft firewall service logs in WC3 format (w3c_fws)
- Web Proxy logs in IIS format (iis_web)
- Microsoft firewall service logs in IIS format (iis_fws)

The W3C event format is the preferred event log format. The W3C format contains a standard heading with the version information and all of the fields that are expected in the event payload. You can customize the W3C event format for the firewall service log and the web proxy log to include or exclude fields from the event logs.

Most administrators can use the default W3C format fields. If the W3C format is customized, the following fields are required to properly categorize events:

*Table 35. W3C format required fields*

| Required field | Description |
|---|---|
| Client IP (c-ip) | The source IP address. |
| Action | Action that is taken by the firewall. |
| Destination IP (r-ip) | The destination IP address. |
| Protocol (cs-protocol) | The application protocol name, for example, HTTP or FTP. |
| Client user name (cs-username) | The User account that made the data request of the firewall service. |
| Client user name (username) | The User account that made the data request of the web proxy service. |

## Microsoft ISA directory structure for event collection

The event logs that are monitored by WinCollect are defined by the root directory that you configure in your log source.

When you specify a root log directory, WinCollect evaluates the directory folder and recursively searches the subfolders to determine when new events are written to the event log. By default, the WinCollect plug-in for Microsoft ISA polls the root log directory for updated event logs every 5 seconds.

| *Table 36. Event log default directory structure for Microsoft ISA* | |
|---|---|
| **Version** | **Root Log Directory** |
| Microsoft ISA 2006 | `%systemroot%\LogFiles\IAS\` |
| Microsoft Threat Management Gateway | `<Program Files>\<Forefront Directory>\ISALogs\` |

## Microsoft ISA protocol parameters

| *Table 37. Microsoft ISA protocol parameters* | |
|---|---|
| **Parameter** | **Description** |
| **Log Source Type** | **Microsoft ISA** |
| **Protocol Configuration** | **WinCollect Microsoft ISA / Forefront TMG** |
| **Local System** | To collect local events, the WinCollect agent must be installed on the same host as your Microsoft ISA or Forefront TMG server. The log source uses local system credentials to collect and forward events to QRadar. |
| **Root Directory** | When you specify a remote file path, use a dollar sign, $, instead of a colon, :, to represent your drive name.<br>Microsoft ISA 2006<br>• For a local directory path, use `%systemroot%\LogFiles\ISA\`<br>• For a remote directory path, use `\<ISA server IP>\%systemroot%\LogFiles\ISA\`<br>Microsoft Threat Management Gateway<br>• For a local directory path, use <*Program Files*>\<*Forefront Directory*>\ISALogs\<br>• For a remote directory path, use \\<*ISA server IP*>\<*Program Files*>\<*Forefront Directory*>\ISALogs\ |
| **File Monitor Policy** | The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your event log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote event log and compares the file to the last polling interval. If the event log contains new events, the event log is retrieved. |
| **Polling Interval** | The amount of time between queries to the root log directory for new events. |

**Related reference**

Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

## Juniper Steel-Belted Radius log source configuration options

Use the reference information to configure the WinCollect plug-in for Juniper Steel-Belted Radius.

*Table 38. Juniper Steel-Belted Radius protocol parameters*

| Parameter | Description |
|---|---|
| **Log Source Type** | **Juniper Steel-Belted Radius** |
| **Protocol Configuration** | **WinCollect Juniper SBR** |
| **Local System** | To collect local events, the WinCollect agent must be installed on the same host as the Juniper Steel-Belted Radius server. The log source uses local system credentials to collect and forward events to QRadar. |
| **Root Directory** | The directory that contains the files that you want to monitor. The QRadar user interface does not verify the path to the root directory. Ensure that you enter a valid local Windows path. |
| **File Monitor Policy** | The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your event log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote event log and compares the file to the last polling interval. If the event log contains new events, the event log is retrieved. |
| **Polling Interval** | The amount of time between queries to the root log directory for new events. |

## Microsoft SQL Server log source configuration options

Use the reference information to configure the WinCollect plug-in for Microsoft SQL Server.

### Microsoft SQL Server Error Logs

The error log is a standard text file that contains Microsoft SQL Server information and error messages. WinCollect monitors the error log for new events and forwards the event to IBM Security QRadar. The error log provides meaningful information to assist you in troubleshooting issues or alerting you to potential or existing problems. The error log output includes the time and date the message was logged, the source of the message, and the description of the message. If an error occurs, the log contains the error message number and a description. Microsoft SQL Servers retain backups of the last six error log files.

WinCollect can collect Microsoft SQL server error log events. To collect Microsoft SQL Server audit and authentication events, you configure the Microsoft SQL Server DSM. For more information, see the *IBM Security QRadar DSM Configuration Guide*.

WinCollect agents support local collection and remote polling for Microsoft SQL Server installations. To remotely poll for Microsoft SQL Server events, you must provide administrator credentials or domain administrator credentials. If your network policy restricts the use of administrator credentials, you can

install a WinCollect agent on the same host as your Microsoft SQL Server. Local installations of WinCollect do not require special credentials to forward events to QRadar.

The Microsoft SQL Server event logs that are monitored by WinCollect are defined by the directory path that you specify in your WinCollect SQL log source. The following table lists the default directory paths for the **Root Log Directory** field in your log source.

*Table 39. Default root log directory paths Microsoft SQL events*

| Microsoft SQL version | Collection type | Root log directory |
|---|---|---|
| 2012 | Local | `C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG` |
| 2012 | Remote | `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG` |
| 2014 | Local | `C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\LOG` |
| 2014 | Remote | `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\LOG` |
| 2016 | Local | `C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\LOG` |
| 2016 | Remote | `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\LOG` |
| 2017 | Local | `C:\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL14.MSSQLSERVER\MSSQL\LOG` |
| 2017 | Remote | `\\HOSTNAME\C$\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL14.MSSQLSERVER\MSSQL\LOG` |
| 2019 | Local | `C:\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL15.MSSQLSERVER\MSSQL\LOG` |
| 2019 | Remote | `\\HOSTNAME\C$\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL15.MSSQLSERVER\MSSQL\LOG` |

Log files that do not match the SQL event log format are not parsed or forwarded to QRadar.

## Supported versions of Microsoft SQL Server

The WinCollect plug-in for Microsoft SQL server supports the following Microsoft SQL software versions:

- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016
- Microsoft SQL Server 2017
- Microsoft SQL Server 2019

The following table describes the Microsoft SQL server protocol parameters.

*Table 40. Microsoft SQL Server protocol parameters*

| Parameter | Description |
|---|---|
| **Log Source Type** | **Microsoft SQL** |
| **Protocol Configuration** | **WinCollect Microsoft SQL** |

| *Table 40. Microsoft SQL Server protocol parameters (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Root Directory** | Microsoft SQL 2012<br>• For a local directory path, use `C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Log`<br>• For a remote directory path, use `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Log`<br>Microsoft SQL 2014<br>• For a local directory path, use `C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Log`<br>• For a remote directory path, use `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Log`<br>Microsoft SQL 2016<br>• For a local directory path, use `C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\LOG`<br>• For a remote directory path, use `\\SQL IP address\c$\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Log`<br>Microsoft SQL 2017<br>• For a local directory path, use `C:\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL14.MSSQLSERVER\MSSQL\LOG`<br>• For a remote directory path, use `\ \HOSTNAME\C$\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL14.MSSQLSERVER\MSSQL\LOG`<br>Microsoft SQL 2019<br>• For a local directory path, use `C:\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL15.MSSQLSERVER\MSSQL\LOG`<br>• For a remote directory path, use `\ \HOSTNAME\C$\PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL15.MSSQLSERVER\MSSQL\LOG` |
| **File Monitor Policy** | The **Notification-based (local)** option uses the Windows file system notifications to detect changes to your event log.<br>The **Polling-based (remote)** option monitors changes to remote files and directories. The agent polls the remote event log and compares the file to the last polling interval. If the event log contains new events, the event log is retrieved. |

**Related reference**

Windows log source parameters

Common parameters are used when you configure a log source for a WinCollect agent or a WinCollect plug-in. Each WinCollect plug-in also has a unique set of configuration options.

## NetApp Data ONTAP configuration options

Use this reference information to configure the WinCollect plug-in for NetApp ONTAP.

| Table 41. NetApp Data ONTAP parameters | |
|---|---|
| **Parameter** | **Description** |
| Log Source Type | NetApp Data ONTAP |
| Protocol Configuration | WinCollect NetApp Data ONTAP |
| User Name | The account name that is used to log in to the Windows domain or system. |
| Domain | The network domain to which the user name belongs. |
| Target Directory | The network path to the directory where you want to monitor files. This path is not verified by IBM Security QRadar user interface. Ensure that you type a valid Windows UNC path that is shared by the NetApp appliance. |
| Polling Interval | The amount of time between queries to the remote directory for new event log files. Even though the remote device does not generate new files in a period of less than 60 seconds, the optimal polling interval is less than 60 seconds. This practice ensures that the collection of files resumes when WinCollect is restarted. |
| WinCollect Agent | The WinCollect Agent that you want to use to collect NetApp Data ONTAP events. |

### Version and file type support

Versions:

- NetApp Data ONTAP 8.x
- NetApp Data ONTAP 9.x

Filetype: Windows Event Log (EVT) and (EVTX)

## Configuring a TLS log source

To encrypt events and send to QRadar, you must configure a log source with a TLS Syslog protocol to establish communication with QRadar on port 6514.

### Procedure

1. Log in to QRadar.
2. Click the **Admin** tab.
3. On the navigation menu, click **Data Sources**.
4. Click **Log Sources** > **Add**.
5. Configure the following parameters:

| Table 42. TLS log source for Wincollect destination | |
|---|---|
| **Parameter** | **Description** |
| **Protocol Configuration** | **TLS Syslog** |
| **Log Source Identifier** | An IP address or host name to identify the log source. |
| **TLS Listen Port** | The default TLS listen port is 6514. |

| *Table 42. TLS log source for Wincollect destination (continued)* | |
|---|---|
| **Parameter** | **Description** |
| **Authentication Mode** | The mode by which your TLS connection is authenticated. If you select the **TLS and Client Authentication** option, you must configure the certificate parameters. |
| **Client Certificate Path** | The absolute path to the client-certificate on disk. The certificate must be stored on the QRadar Console or Event Collector for this log source. |
| **Certificate Type** | The type of certificate to use for authentication for the server certificate and server key.<br>Select one of the following options from the **Certificate Type** list:<br>• **Generated Certificate**<br>• **Single Certificate and Private Key**<br>• **PKCS12 Certificate and Password** |
| **Generated Certificate** | This option is available when you configure the **Certificate Type**.<br>If you want to use the default certificate and key that is generated by QRadar for the server certificate and server key, select this option. |
| **Single Certificate and Private Key** | This option is available when you configure the **Certificate Type**.<br>If you want to use a single PEM certificate for the server certificate, select this option and then configure the following parameters:<br>• **Provided Server Certificate Path** - The absolute path to the server certificate.<br>• **Provided Private Key Path** - The absolute path to the private key.<br>**Note:** The corresponding private key must be a DER-encoded PKCS8 key. The configuration fails with any other key format. |
| **PKCS12 Certificate and Password** | This option is available when you configure the **Certificate Type**.<br>If you want to use a PKCS12 file that contains the server certificate and server key, select this option and then configure the following parameters:<br>• **PKCS12 Certificate Path** - Type the file path for the PKCS12 file that contains the server certificate and server key.<br>• **PKCS12 Password** - Type the password to access the PKCS12 file.<br>• **Certificate Alias** - If there is more than one entry in the PKCS12 file, an alias must be provided to specify which entry to use. If there is only one alias in the PKCS12 file, leave this field blank. |
| **Max Payload Length** | The maximum payload length (characters) that is displayed for TLS Syslog message. |

*Table 42. TLS log source for Wincollect destination (continued)*

| Parameter | Description |
|---|---|
| **Maximum Connections** | The **Maximum Connections** parameter controls how many simultaneous connections the TLS Syslog protocol can accept for each Event Collector. There is a limit of 1000 connections across all TLS syslog log source configurations for each Event Collector. The default for each device connection is 50.<br>**Note:** Automatically discovered log sources that share a listener with another log source. For example, if you use the same port on the same event collector, it counts only one time towards the limit. |
| **TLS Protocols** | The TLS Protocol to be used by the log source. Select one of the following options:<br>• TLS 1.2 and above<br>• TLS 1.1 and above<br>• TLS 1.0 and above<br>To avoid security vulnerabilities, use TLS 1.2 and above. |
| **Use As A Gateway Logsource** | Sends collected events through the QRadar Traffic Analysis Engine to automatically detect the appropriate log source.<br>You must select this in order for QRadar to detect/create the correct log source for events.<br>When this option is not selected and **Log Source Identifier Pattern** is not configured, QRadar receives events as unknown generic log sources. |

*Table 42. TLS log source for Wincollect destination (continued)*

| Parameter | Description |
|---|---|
| **Log Source Identifier Pattern** | If you selected **Use As A Gateway Log Source**, use this option to define a custom log source identifier for events that are being processed and for log sources to be automatically discovered when applicable. If you don't configure the **Log Source Identifier Pattern**, QRadar receives events as unknown generic log sources.<br>Use key-value pairs to define the custom Log Source Identifier. The key is the Identifier Format String, which is the resulting source or origin value. The value is the associated regex pattern that is used to evaluate the current payload. This value also supports capture groups that can be used to further customize the key.<br>Define multiple key-value pairs by typing each pattern on a new line. Multiple patterns are evaluated in the order that they are listed. When a match is found, a custom Log Source Identifier displays.<br>The following examples show multiple key-value pair functions.<br>**Patterns**<br>`VPC=\sREJECT\sFAILURE`<br>`$1=\s(REJECT)\sOK`<br>`VPC-$1-$2=\s(ACCEPT)\s(OK)`<br>**Events**<br>`{LogStreamName: LogStreamTest,Timestamp: 0,Message: ACCEPT OK,IngestionTime: 0,EventId: 0}`<br>**Resulting custom log source identifier**<br>VPC-ACCEPT-OK |
| **Enable Multiline** | Aggregate multiple messages into single events based on a Start/End Matching or an ID-Linked regular expression. |
| **Aggregation Method** | This parameter is available when **Enable Multiline** is turned on.<br>• **ID-Linked** - Processes event logs that contain a common value at the beginning of each line.<br>• **Start/End Matching** - Aggregates events based on a start or end regular expression (regex). |
| **Event Start Pattern** | This parameter is available when **Enable Multiline** is turned on and the **Aggregation Method** is set to **Start/End Matching**.<br>The regular expression (regex) that is required to identify the start of a TCP multiline event payload. Syslog headers typically begin with a date or timestamp. The protocol can create a single-line event that is based on solely on an event start pattern, such as a timestamp. When only a start pattern is available, the protocol captures all the information between each start value to create a valid event. |

| Table 42. TLS log source for Wincollect destination (continued) | |
|---|---|
| **Parameter** | **Description** |
| **Event End Pattern** | This parameter is available when **Enable Multiline** is turned on and the **Aggregation Method** is set to **Start/End Matching**.<br>This regular expression (regex) that is required to identify the end of a TCP multiline event payload. If the syslog event ends with the same value, you can use a regular expression to determine the end of an event. The protocol can capture events that are based on solely on an event end pattern. When only an end pattern is available, the protocol captures all the information between each end value to create a valid event. |
| **Message ID Pattern** | This parameter is available when **Enable Multiline** is turned on and the **Aggregation Method** is set to **id-Linked**.<br>This regular expression (regex) required to filter the event payload messages. The TCP multiline event messages must contain a common identifying value that repeats on each line of the event message. |
| **Time Limit** | This parameter is available when **Enable Multiline** is turned on and the **Aggregation Method** is set to **id-Linked**.<br>The number of seconds to wait for more matching payloads before the event is pushed into the event pipeline. The default is 10 seconds. |
| **Retain Entire Lines during Event Aggregation** | This parameter is available when **Enable Multiline** is turned on and the **Aggregation Method** is set to **id-Linked**.<br>If you set the **Aggregation Method** parameter to **ID-Linked**, you can enable **Retain Entire Lines during Event Aggregation** to discard or keep the part of the events that comes before **Message ID Pattern** when concatenating events with the same ID pattern together. |
| **Flatten Multiline Events Into Single Line** | This parameter is available when **Enable Multiline** is turned on.<br>Shows an event in one single line or multiple lines. |
| **Event Formatter** | This parameter is available when **Enable Multiline** is turned on.<br>Use the **Windows Multiline** option for multiline events that are formatted specifically for Windows. |

6. Click **Save**.

## Creating a TLS log source destination for managed agents

Create a TLS destination if you want to send encrypted events to IBM Security QRadar appliances. For any existing log sources that are using WinCollect you must ensure that they use the TLS destination you created so that the events are encrypted.

### Procedure

1. Click the **Admin** tab.
2. Create a TLS log source destination.

a) Click **Data Sources** > **WinCollect**.

b) In the **WinCollect** window, click **Destinations** > **Add**.

c) Give the destination a name, and specify the IP address or hostname of the console.

d) In the **Protocol** menu, select **TCP/TLS (Encrypted)**.

e) Paste the certificate, including the BEGIN and END lines.

Find the self-signed certificate in `/opt/qradar/conf/trusted_certificates/syslog-tls.cert`.

f) Click **Save**.

3. Create a TLS Syslog log source where the log source type is **Universal DSM** and the protocol type is **TLS Syslog**.

For more information about adding a log source, see Adding a log source to receive events (https://www.ibm.com/docs/en/qradar-common?topic=app-adding-log-source-receive-events).

# Adding a log source to a WinCollect agent

When you add a new log source to a WinCollect agent or edit the parameters of a log source, the WinCollect service is restarted. The events are cached while the WinCollect service restarts on the agent.

## Before you begin

If you configure a log source that uses a WinCollect plug-in, you must read the requirements and prepare the third-party device. For more information, see WinCollect plug-in requirements.

## Procedure

1. Click the **Admin** tab.
2. On the navigation menu, click **Data Sources**.
3. Click the **WinCollect** icon.
4. Click **Agents**.
5. Select the WinCollect agent, and click **Log Sources** and then click **Add**.
6. Choose one of the following options:
   - For a WinCollect log source, select **Microsoft Windows Security Event Log** from the **Log Source Type** list and then select WinCollect from the **Protocol Configuration** list.
   - For a WinCollect plug-in select the WinCollect plug-in option from the **Log Source Type** list, and then configure the specific parameters. For information about these parameters, see the configuration options for log sources that use WinCollect plug-ins.
7. Configure the generic log source parameters.
8. Click **Save**.
9. On the **Admin** tab, click **Deploy Changes**.

# Bulk log sources for remote event collection

Bulk log sources are designed for systems that have multiple log sources with the same protocol configuration.

## Procedure

1. Create a destination for Windows events on each IBM QRadar appliance that you want to use for Windows event collection. See “Adding a destination” on page 37.

   **Important:** It is helpful to provide a destination name that includes the IP address, such as "Agent1_1.2.3.4". If you have to edit the log source and change a destination in the future, you can

determine the IP address for the destination. Also, set the throttle value to 5000 EPS, which is the max EPS rate for a WinCollect agent.

2. Create bulk log sources. See "Adding log sources in bulk for remote collection" on page 88.
3. Wait for the configurations to be pushed to the remote agents.
4. Verify in the **Log Activity** tab that events being received.

# Adding log sources in bulk for remote collection

You can add multiple log sources at one time in bulk to IBM QRadar. The log sources must share a common configuration protocol and be associated with the same WinCollect agent.

You can upload a text file that contains a list of IP addresses or host names, run a query against a domain controller to get a list of hosts, or manually enter a list of IP addresses or host names by typing them in one at a time.

Depending on the number of WinCollect log sources that you add at one time, it can take time for the WinCollect agent to access and collect all Windows events from the log source list.

## Before you begin

Ensure that you created destinations so that WinCollect agents can send Windows events to QRadar appliances. Ensure that you created one destination for each QRadar Event Collector 16xx or 18xx appliance.

Plan your bulk collection strategy with the WinCollect Event Log Report tool. For more information, see GitHub (https://github.com/ibm-security-intelligence/wincollect).

## About this task

You can have a maximum of 500 log sources for each managed WinCollect agent. You must also remain under 5,000 EPS for local collection and 2,500 EPS for remote polling on the WinCollect Agent. You can review the Event Viewer on the Windows systems to determine how many EPS are generated in each hour. Divide that value by 3600 seconds to get the EPS rate. This calculation helps you to plan how many agents you need to install. Alternately, look at events over a 24-hour period to see how busy each Windows server is. This helps determine how to tune agents and avoid minimum and maximum EPS rates that you see only when reviewing hour-by-hour.

## Procedure

1. On the **Admin** tab navigation menu, click **Data Sources**, and then click the **WinCollect** icon.
2. Select the WinCollect agent that you want to assign log sources to, and click **Log Sources**.
3. Click **Bulk Actions** > **Bulk Add**.
4. Provide a name for the bulk log source. To make it easy to locate, specify the name as the WinCollect agent that does remote collection.
5. From the **Log Source Type list** box, select **Microsoft Windows Security Event Log**.
6. From the **Protocol Configuration list** box, select **WinCollect**.
7. Use the tuning value specified by the WinCollect Event Log Report tool to tune your log sources appropriately.
8. Select all of the **Standard Log Types** check boxes. The WinCollect agent reads and forwards these remote logs to QRadar.

   **Important:** Do not select **Forwarded Events** the check box. Forwarded events is a special use case. Selecting this option will not add multiple log sources correctly.
9. Select all of the **Event Types** check boxes.
10. Select the **Enable Active Directory Lookups** check box. This option identifies user names in Windows events that appear as a hexadecimal and resolves them to human readable user names.
11. From the **WinCollect Agent** list, select the Windows host that manages the log source.

12. From the **Target Internal Destination** list, select the QRadar appliance that receives and processes the Windows events.
13. Add the IP addresses for the Windows operating systems that you want to remotely poll for events.

    You can upload a text file that contains a list of IP addresses or host names, run a query against a domain controller to get a list of hosts, or manually enter a list of IP addresses or host names by typing them in one at a time.

    Depending on the number of WinCollect log sources that you add at one time, it can take time for the WinCollect agent to access and collect all Windows events from the log source list.
14. Click **Save** and then click **Continue**.

## What to do next

Wait for the configurations to be pushed to the remote agents. Verify in the **Log Activity** tab that events are received.

**Related tasks**

Adding a log source to a WinCollect agent

When you add a new log source to a WinCollect agent or edit the parameters of a log source, the WinCollect service is restarted. The events are cached while the WinCollect service restarts on the agent.

# Chapter 6. Troubleshooting WinCollect deployment issues

If you experience issues with your WinCollect deployment, the following information might help you identify and resolve the issues.

In a complex WinCollect deployment with many assets, identifying the source and cause of problems can be difficult. Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and explain how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the WinCollect Support representative know where to start to find the cause of the problem. This step includes asking some basic questions:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, and that is the best way to start down the path of problem resolution. After you have a clear description, you can investigate the cause of and solution to the problem, or contact WinCollect Support to assist you in the investigation.

### Where to get help

Troubleshooting is not the same as problem solving, although during the process of troubleshooting, you can often obtain enough information to solve a problem. However, sometimes you might encounter a problem that you cannot solve by yourself, even after you determine its cause. If you are unable to solve a problem on your own, you can contact WinCollect Support for a solution.

You can also use any of the following resources to help you find a solution to your problem:

- The WinCollect 101 Community (https://www.ibm.com/community/qradar/home/wincollect/)
- The WinCollect Forums (http://ibm.biz/wincollectforums)

## Common problems

The following topics describe some known problems that can occur in a WinCollect deployment, and provide solutions to those problems.

If your problem is described here, you can try these solutions before you need to contact support.

## Replacing the default certificate in QRadar generates invalid PEM errors

Replacing the default certificate in QRadar causes the `ConfigurationServer.PEM` file to change, affecting all WinCollect agents in the deployment. To fix this issue, you must replace the `ConfigurationServer.PEM` file on the Windows host.

### About this task

WinCollect agents receive rejection messages because the incorrect certificate is passed when the agents attempt to communicate with the updated QRadar appliance. The following error message appears in the logs:

```
May 17 17:06:31 ::ffff:IP ADDRESS [ecs-ec] [WinCollectConfigHandler_4]
com.q1labs.sem.semsources. wincollectconfigserver.WinCollectConfigHandler: [ERROR]
```

```
[NOT:0000003000] [192.0.2.0/- -] [-/- -]Agent with ip: IP ADDRESS tried to connect
with an invalid PEM
```

The IP address of the agent that is attempting to communicate is displayed. The WinCollect agent also sends LEEF Syslog messages to inform the administrator of the communication issue due to the invalid certificate. To fix this issue, you must replace the `ConfigurationServer.PEM file` on the Windows host.

**Note:** This action must be completed by a Windows administrator or a user that has privileges to delete files from the remote Windows host.

### Procedure

1. Open a remote desktop connection to the WinCollect Agent that is unable to communicate.
2. Click **Start** > **Run**.
3. Type `services.msc`, then click **OK**.
4. Stop the **WinCollect** service.
5. On the Windows host, navigate to the WinCollect configuration folder.
   By default, the folder path is: `C:\ProgramFiles\IBM\WinCollect\config`
6. Delete `ConfigurationServer.PEM`.
7. From the **Services** window, start the **WinCollect** service.

### Results

After the WinCollect service restarts, the agent attempts to contact the QRadar appliance that manages the Windows host. The QRadar appliance detects the missing `ConfigurationServer.PEM` file and issues a replacement against the existing certificate. This practice replaces the old file with a new `ConfigurationServer.PEM` file that includes the updated certificate.

## The Statistics Subsystem

The Statistics Subsystem collects events per second (EPS) data from all log sources and destinations in a single text file.

The system creates the `logs\Statistics.txt` file and populates it with collected EPS statistics every 5 minutes.

When the Agent starts, it writes the first set of collected statistics to the end of the `Statistics.txt` file, leaving older statistics intact. The system then writes the content at the same location with new statistics every 5 minutes.

You can change the interval at which new statistics are reported in the `logconfig.xml` file. The `ReportEvery` parameter specifies the number of minutes between each report. The default value is 5 minutes.

## Event ID 1003 splits the message in QRadar

Windows Event ID 1003 can exceed the default maximum payload size in QRadar. It is then split into two separate messages.

### About this task

The default maximum payload size in QRadar is 4096 bytes. If Event ID 1003 messages are being split, you must increase the maximum payload size to keep the messages intact.

Follow these steps to increase the maximum payload size.

### Procedure

1. Log in to the Console as an administrator.

2. Click the **Admin** tab.
3. Click **System Settings** > **Advanced**.
4. On the **System Settings** pane, update the **Max TCP Syslog Payload Length** value to 8,192.

   **Tip:** Extremely large payload values can impact performance of the event pipeline. Do not increase the TCP Payload Length Value above 8,192 bytes without contacting IBM support.
5. Click **Save**.
6. On the **Admin** tab, click **Advanced** > **Deploy Full Configuration**.

   **Important:** Completing a full deployment restarts all services on all QRadar appliances. Verify whether reports are running before you run the deployment, as a full deployment stops reports that are in progress. These reports must be manually restarted by a user or the administrator. This procedure also temporarily stops event and flow collection on all appliances while services are restarting. To avoid these issues, make this change during a maintenance window.
7. Click **Continue** to start the full deployment process.

### Results

After the deployment completes, all QRadar managed hosts are sent the change to accept larger TCP payload length. The payloads across all managed hosts do not truncate the event message, unless they exceed 8,192 bytes.

## WinCollect files are not restored during a configuration restore.

When you complete a configuration restore and some WinCollect files are not restored, it might be because the installation ISO contains a previous version of WinCollect.

The QRadar ISO contains a built-in version of WinCollect. When you restore by using that ISO, it deploys the WinCollect files that are stored in that ISO, rather than the files from your backup.

To remedy this issue, you must install the WinCollect SFS that matches the version of WinCollect in your backup before you restore the configuration. Perform the following tasks in this order:

1. Perform QRadar backup.
2. Bring new hardware online and deploy the ISO.
3. Install the WinCollect SFS that matches the version of WinCollect in your backup on the Console.
4. Restore the configuration backup.

The appropriate WinCollect files are deployed with the configuration restore.

## Windows 10 (1803) can't read the Security Bookmark file

Log sources for Windows 10, build 1803 fail to read the Security Bookmark file after the host is restarted.

This is a known issue with Windows 10, build 1803. After you install WinCollect and restart the computer, the log source can fail to read the Security Bookmark file.

To fix this issue on WinCollect 7.2.5, edit any log sources that are experiencing the issue with an XPATH that includes the Security event log and any other channels that you're monitoring.

To fix this issue on WinCollect 7.2.6 or later, edit the log source to use MSEVEN6.

## Resolving log source error after WinCollect update

An error message might appear when you attempt to edit a log source after you upgrade WinCollect, IBM QRadar, a Device Support Module (DSM), a protocol, or any Vulnerability Information Services (VIS)

components. To remove cached files, restart the QRadar web service and clear the QRadar files from your browser cache.

### Before you begin

You must have SSH access and root account credentials.

### About this task

The following message indicates that the web server didn't restart after QRadar was updated:

```
An error has occurred. Refresh your browser (press F5) and attempt the action again.
If the problem persists, please contact customer support for assistance.
```

A file might be cached by QRadar web service or your desktop browser. You must restart QRadar web service and remove the cached files on your desktop.

### Procedure

1. Use SSH to log in QRadar.
2. Stop the QRadar web service by typing the following command:

   ```
   service tomcat stop
   ```
3. Keep one web browser window open.
4. To clear your browser cache, go to your web browser's preference settings.
5. Restart the browser.
6. Restart the QRadar web service by typing the following command:

   ```
   service tomcat start
   ```

## WinCollect log file

The WinCollect log file provides information about your deployment. Logs provide valuable information for troubleshooting issues.

### WinCollect log overview

WinCollect generates log event extended format (LEEF) messages during installation and configuration and writes them to a single log file. The server in the **Status Server** field receives the LEEF messages through the syslog. These messages report on the status of the WinCollect service, authorization token, configuration, and more.

**Example:**

The following example displays a LEEF message that alerts administrators that the WinCollect agent is generating more events than the log source is tuned for.

```
<13>Sep 22
09:07:56 IPADDRESS LEEF:1.0|IBM|WinCollect|7.2|3|src=MyHost.example.com
dst=10.10.10.10
sev=4 log=Device.WindowsLog.EventLog.MyHost.example.com.System.Read
msg=Reopening event log
due to falling too far behind (approx 165 logs skipped). Incoming
EPS r.avg/max =
150.50/200.00. Approx EPS possible with current tuning = 40.00
```

For more information, see Log Source Event Rates and Tuning Profiles (http://www.ibm.com/support/docview.wss?uid=swg21672193).

You search for syslog messages by using the IP address of the WinCollect agent. QRadar tracks information from the audit log to determine when log sources are created, when searches are run, and so on.

## WinCollect log types

The default log directory is `C:\Program Files\IBM\WinCollect\logs\`. The log file is named `WinCollect.log`.

Each log entry is tagged with an identifier that indicates the entry type:

- System
- Code
- Device

```
03-08 11:08:16.306 INFO  Code.PayloadRouter : Using 3 router threads.
03-08 11:08:16.306 INFO  Code.PayloadRouter : Using stats sweep period of 30 seconds.
03-08 11:08:16.306 INFO  System.ComponentFactory : Service PayloadRouter v7.2.8 initialized
03-08 11:08:16.306 INFO  Device.Windows2008EventCollector : Windows2008 Event Collector 7.2.8.58 initialized
03-08 11:08:16.306 INFO  System.ComponentFactory : Service Windows2008EventCollector v7.2.8 initialized
03-08 11:08:16.306 INFO  Device.Service.FileForwarderDevice : Initializing FileForwarder Device Service...
03-08 11:08:16.306 INFO  Device.Service.FileForwarderDevice : FileForwarder Device Service initialized.
```

The following table describes the types of log entries in the WinCollect log file.

*Table 43. WinCollect log entry types*

| Log Entry Type | Description |
|---|---|
| System | Indicates system information, such as the operating system that the agent is installed on, RAM and CPU information from the operating system, service start-up information, and WinCollect version information. |
| Code | Indicates information about spillover and cache messages, file reader messages, authorization token messages, IP address or host name information for the local host, issues with destinations, log source auto-creation, stand-alone mode messages, and thread or process start-up and shutdown messages. Use these entries to investigate the WinCollect configuration. Code entries do not provide information about event collection. |

| *Table 43. WinCollect log entry types (continued)* | |
|---|---|
| **Log Entry Type** | **Description** |
| Device | Created when WinCollect collects events, the protocols that run event log collection. The following issues are logged as Device entries:<br>Loading Plug-in<br>Connection issues<br>Permission or Authentication<br>Windows error codes (hex value codes provided by the operating system, such as 0x000005 access denied)<br>File path or location<br>Event log is overdue to be polled<br>Event log transactions<br>RPC is unavailable (unable to find the location that you specified)<br>Reopening due to falling too far behind (tuning messages) |

### Disk space management for log files

WinCollect manages disk space for logs by generating a "_1" version when the log size exceeds 20 MB. After a "_5" version is created, WinCollect deletes the oldest version of the log.

WinCollect also manages disk space by archiving checkpoint folders. When QRadar updates WinCollect with new code, the checkpoint folders store a backup of the replaced code. WinCollect archives the oldest patch checkpoint folder after 10 are created. WinCollect creates an archive folder that contains a list of files in the patch checkpoint folder, and a compressed file of the AgentConfig.xml file. WinCollect then deletes the patch checkpoint folder that it archived.

## InfoX debug logs

InfoX debug logs make debugging WinCollect easier, without interfering with performance.

By default, InfoX is enabled and logs events for the first five minutes that the agent runs, for a maximum of 5,000 log entries. After that, InfoX logs events for one minute every 15 minutes, for a maximum of 200 log entries. InfoX generates debug logs even if your log level is set to **`info`**.

You can edit the InfoX configuration by adding any of these parameters to the `logconfig.xml` file.

| *Table 44. InfoX configuration options* | |
|---|---|
| **Parameter** | **Description** |
| InfoX.enabled | Used to enable or disable InfoX.<br>Example: `InfoX.enabled=true` |
| InfoX.startLen | The number of seconds to run the agent at startup. To disable this feature, set this value to `0`.<br>Example: `InfoX.startLen=300` |

| *Table 44. InfoX configuration options (continued)* | |
|---|---|
| **Parameter** | **Description** |
| InfoX.startMax | The maximum number of events that can be logged at startup.<br>Example: `InfoX.startMax=5000` |
| InfoX.nextWait | The number of seconds to wait for the next logging period.<br>Example: `InfoX.nextWait=900` |
| InfoX.nextLen | The number of seconds to run the agent at each interval. To disable this feature, set this value to `0`.<br>Example: `InfoX.nextLen=60` |
| InfoX.nextMax | The maximum number of events that can be logged at each interval.<br>Example: `InfoX.nextMax=200` |

## WinCollect not supported by Data Synchronization app

The officially supported QRadar disaster and recovery solution, the QRadar Data Synchronization app, currently does not support WinCollect agent configuration updates.

For the uninterrupted flow of events, you can configure a secondary destination to act as a failover when the primary destination is unreachable. You can also configure the time interval for the agent to re-check if the primary destination is still unreachable. After the agent is able to connect, it will revert to the primary destination. If both the primary and secondary destinations are unreachable, or if no secondary destination is configured, the events are cached by the agent on disk until a destination is reachable. At that point, the agent continues to send events at the set EPS rate until all the cached events are consumed.

**Related tasks**

“Adding a secondary destination” on page 38
You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
170 Tracer Lane,
Waltham MA 02451, USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.


Microsoft, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.