Juniper Networks | Engineering Simplicity

Juniper Secure Analytics WinCollect 10 User Guide

Published
2023-07-25

RELEASE
7.5.0

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Secure Analytics WinCollect 10 User Guide
7.5.0

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide

Use this guide to understand how to can use JSA to manage and collect Windows-based events.

1

CHAPTER

WinCollect 10 Overview

WinCollect 10 overview

IN THIS SECTION

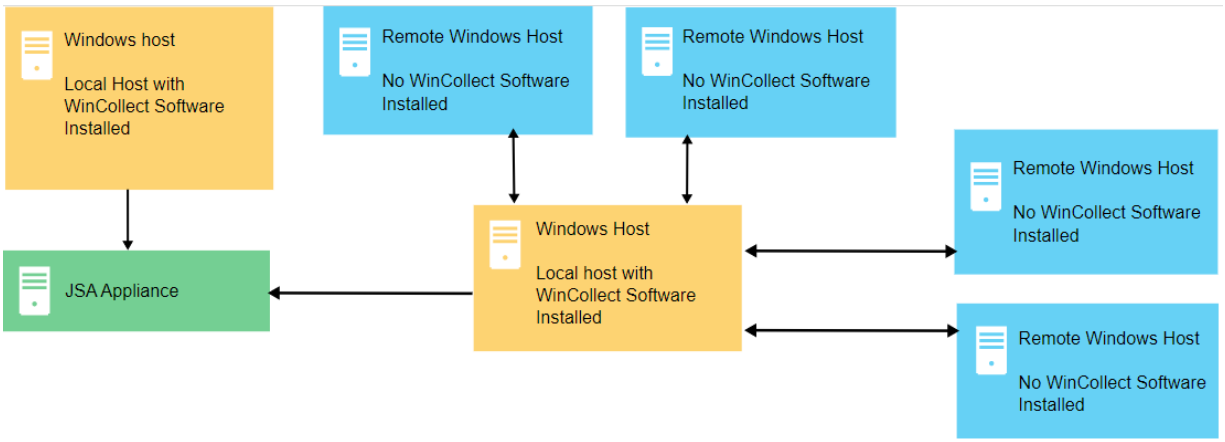
WinCollect is a Syslog event forwarder that administrators can use to forward events from Windows logs to JSA. WinCollect can collect events from systems locally or be configured to remotely poll other Windows systems for events.

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

NOTE: You can install WinCollect 10 as a stand-alone agent only.

In a stand-alone deployment, the WinCollect software is installed on a Windows host that is not managed through JSA to control the log sources. There are no performance differences between a managed and stand-alone agent. The agent can gather events from itself (local), connect to a remote Windows endpoint to collect events (remote), or both. The agent then sends both the local and remote events to your JSA deployment.

Figure 1: 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 JSA.

Stand-alone WinCollect mode has the following capabilities:

- Configure each WinCollect agent by using the WinCollect 10 Console.
- 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 JSA over TLS Syslog.
- Automatically create a local source at the time of agent installation.

What's new in WinCollect 10

IN THIS SECTION

WinCollect 10 is a major new release for JSA. This release is available now for stand-alone deployments.

Significant performance improvements

- Across many different use cases (high and low eps, high eps remote polling), approximately 70-100% improvement in CPU usage and 53-67% improvement in memory usage.

- Increased EPS limitation from 5,000 to 10,000 for local collection.

Installation improvements

Quick Installation	Using only the IP/Hostname of the JSA host, you can have an Agent up and running in seconds, collecting standard Application, System, and Security events.
Installation with script	• No longer requires a paragraph cmd line to install an agent. The installer can now reference an installation script. • The installer uses the configuration in the script to add the sources you want as part of the installation. You are no longer limited to configuring the Windows event log collection as part of the installation. • You can configure any devices that are supported by WinCollect during the installation.
Lightweight installation	~4 MB installation versus 40 MB (installer + patch installer, if needed).

Automatic tuning

- You no longer need to configure the polling interval or guess which tuning profile to use. The WinCollect agent now tunes itself by Source to poll more often when required and less often when the EPS is low.
- Configure which sources you want to use, and let the agent handle the collection of events.

Web-based agent management

Web-based agent management is an optional component for all Agent installations and no longer requires a separate installation as it did with WinCollect 7. Agent management is no longer dependent on .NET3.5.

> **TIP**: The agent management UI works on Internet Explorer, Firefox, or Chrome.

In addition to agent management, the UI contains the following features:

- Main Dashboard
  - Top Sources - list of the top 10 sources by EPS
  - Errors - lists recent Agent errors, such as connections to JSA or to a remote source.

  - Historical EPS by source graph

- Add source wizard.
  - Wizard to add local or remote sources one at a time or in bulk.

The UI also contains the following support tools:

- Log Viewer
  - Displays the WinCollect log in real time, so you can filter the log as needed.
- Restart WinCollect service - The following options are available during restart to help troubleshoot an issue:
  - Delete Logs
  - Delete Patch/Staging folder
  - Delete Cached Events
  - Delete Bookmarks
  - Start in Debug Mode
- Collect Support files.
  - Click one button to gather all the required log files to provide to L2/L3 Juniper Customer Support.

Use of sources

WinCollect 10 changes the collection paradigm from the typical JSA log source collection to source collection. For example, in JSA, you specify to collect Windows event logs and select which channels you want to collect. In WinCollect 10, each channel you want to collect from is now referred to as a "source," which provides the agent more flexibility. For example, channels no longer need to be polled at the same time; you can now set polling intervals for each source. Sources also provide the ability to more easily apply updates by using update scripts.

NOTE: The other plug-ins (such as Microsoft SQL Server) are also referred to as sources.

Agent Configuration with update scripts

- WinCollect 10 takes templates to the next level. In WinCollect 7, you could update agents by using templates to make wholesale changes to the configuration. Simple tweaks to an existing configuration were not possible. In WinCollect 10, you can make minor changes to the configuration, and add or subtract sources.

- If you want to change the IP destination, you can create a simple update script that you can push out to all your agents.
- The agent configuration is now much simpler and easier to read. Prior agent configurations that were 200+ lines are now reduced to 10 - 20 lines.

Performance comparison between WinCollect versions

The following tables display the comparisons between WinCollect 7.3 and WinCollect 10 memory and CPU usage across different use cases.

Table 1: Use case - Local collection (System, Application, Security), low EPS (<1)

WinCollect version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Windows 10	2 cores	2 GB	0.6 %	5.5 MB
WinCollect 10 (10.0.1)	Windows 10	2 cores	2 GB	0.0 %	2.8 MB
WinCollect 7 (7.3.0-41)	Server 2016	2 cores	4 GB	0.75 %	9 MB
WinCollect 10 (10.0.1)	Server 2016	2 cores	4 GB	0.0 %	4.1 MB
WinCollect 7 (7.3.0-41)	Server 2019	2 cores	4 GB	0.8%	9 MB
WinCollect 10 (10.0.1)	Server 2019	2 cores	4 GB	0.0 %	3.5 MB
Average memory and CPU improvement				100%	53%

Table 2: Use case - Local collection (System, Application, Security), medium EPS (100)

WinCollect version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Windows 10	2 cores	2 GB	1.5 %	5.5 MB
WinCollect 10 (10.0.1)	Windows 10	2 cores	2 GB	0.21 %	3.0 MB
WinCollect 7 (7.3.0-41)	Server 2016	2 cores	4 GB	0.8 %	9.7 MB
WinCollect 10 (10.0.1)	Server 2016	2 cores	4 GB	0.12 %	4.1 MB

Table 2: Use case - Local collection (System, Application, Security), medium EPS (100) ***(Continued)***

WinCollect version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Server 2019	2 cores	4 GB	1.2%	10.5 MB
WinCollect 10 (10.0.1)	Server 2019	2 cores	4 GB	0.10 %	3.6 MB
Average memory and CPU improvement				88%	57%

Table 3: Use case - Local collection (System, Application, Security), high EPS (2500)

WinCollect version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Windows 10	2 cores	2 GB	22 %	11 MB
WinCollect 10 (10.0.1)	Windows 10	2 cores	2 GB	7.0 %	3.0 MB
WinCollect 7 (7.3.0-41)	Server 2016	2 cores	4 GB	17 %	9.7 MB
WinCollect 10 (10.0.1)	Server 2016	2 cores	4 GB	4.7 %	4.1 MB
WinCollect 7 (7.3.0-41)	Server 2019	2 cores	4 GB	18%	15 MB
WinCollect 10 (10.0.1)	Server 2019	2 cores	4 GB	4.7 %	4.6 MB
Average memory and CPU improvement				71%	67%

Table 4: Remote Collection

WinCollect Version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Server 2019	2 cores	4 GB	1.4%	382 MB
WinCollect 10 (10.0.1)	Server 2019	2 cores	4 GB	0.1%	60 MB
Average memory and CPU improvement				93%	84%

Table 5: Use case - 500 endpoint remote collection (System, Application, Security), high EPS (~5000)

WinCollect Version	OS	CPU	Memory	Average CPU	Memory
WinCollect 7 (7.3.0-41)	Server 2019	2 cores	4 GB	46%	425 MB
WinCollect 10 (10.0.1)	Server 2019	2 cores	4 GB	21%	84 MB
Average memory and CPU improvement				54%	80%

2
CHAPTER

Installing WinCollect 10

Installing WinCollect 10

IN THIS SECTION

You can install a new WinCollect 10 stand-alone agent by using the Quick Installation or Advanced Installation options. You can also upgrade an existing WinCollect 7.3.0 or later stand-alone agent to the latest version of WinCollect 10.

Upgrade existing WinCollect agents

Use one of the following methods to upgrade an existing WinCollect stand-alone agent.

"Upgrading a WinCollect 7 agent to WinCollect 10" on page 15

"Upgrading existing WinCollect 10 agents" on page 17

Install new agents

You can install a new WinCollect 10 stand-alone agent by using the Quick Installation or Advanced Installation options.

Quick installation A quick installation only requires you to set the JSA destination. The installation automatically configures collection of application, system, and security events. You can install WinCollect 10 using the GUI installer or the command line.

"Installing WinCollect 10 using the GUI Quick installation" on page 17

"Installing WinCollect 10 using the command line" on page 18

Advanced installation options Use one of the advanced installation options to run a silent installation or specify an installation script that gives the agent instructions on what to monitor and where to send events.

"Installing WinCollect 10 using the Advanced installer" on page 18

"WinCollect 10 Command line installation advanced examples" on page 19

"WinCollect 10 installation script examples" on page 20

TIP: When you specify an installation location, use the full path and not the relative path.

Hardware and software requirements for the WinCollect 10 host

Ensure that the Windows-based computer that hosts the WinCollect 10 agent meets the minimum hardware and software requirements.

Hardware and virtual machine requirements

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

Table 6: Hardware or VM requirements for local collection by using WinCollect

Requirement	Description
Memory	The WinCollect agent has a small memory footprint. The following numbers were generated on virtual machines (VMs) with two logical cores and 2-4GB of memory. One Event per second (EPS) or less: 3.5 MB 100 EPS or less: 3.6 MB 2,500 EPS or less: 4.6 MB 5,000 EPS or less: 6 MB
Processor	Intel Core i3 or equivalent Systems were tested on VMs with two cores and 2 - 4 GB of memory.
Available processor resources	0-20%, depending on CPU, EPS, and number of endpoints polled. See the following table for examples. Very high EPS rates have a direct effect on the Average CPU used by the WinCollect Agent.
Disk space	20 MB for software, plus up to 300 MB for log files. Up to 6 GB might be required, if you store events to disk.

NOTE: WinCollect CPU load depends on several factors, including the number of events per second that are being processed.

The following table shows resources that are used by WinCollect 10, using the minimum recommended provisioned test environments with various EPS counts.

Table 7: Local Collection - System, Application, Security event logs

Profile	OS	CPU	Memory	Average CPU	Memory
Low EPS (<1)	Windows 10	2 cores	2 GB	0.0%	2.8 MB
Low EPS (<1)	Server 2016	2 cores	4 GB	0.0%	4.1MB
Low EPS (<1)	Server 2019	2 cores	4 GB	0.0%	3.5 MB

Table 7: Local Collection - System, Application, Security event logs *(Continued)*

Profile	OS	CPU	Memory	Average CPU	Memory
Medium EPS (100)	Windows 10	2 cores	2 GB	0.21%	3.0 MB
Medium EPS (100)	Server 2016	2 cores	4 GB	0.12%	4.1 MB
Medium EPS (100)	Server 2019	2 cores	4 GB	0.10%	3.6 MB
High EPS (5000)	Windows 10	2 cores	2 GB	14%	4.7 MB
High EPS (5000)	Server 2016	2 cores	4 GB	8%	6.0 MB
High EPS (5000)	Server 2019	2 cores	4 GB	9%	5.7 MB

Table 8: Local Collection - WEF Collector

WinCollect 10 running on a WEF collector can support up to 10k EPS. When collecting events at this high of an EPS, run the Agent on a dedicated host.

Profile	OS	CPU	Memory	Average CPU	Memory
WEF Collector	Server 2019	6 cores	16 GB	4.5%	13 MB

Table 9: Remote Collection

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.

Profile	OS	CPU	Memory	EPS	Endpoints polled	Average CPU	Memory
High EPS / Low Device Count	Server 2019	8 cores	16 GB	5000	10	7.5%	11 MB

Table 9: Remote Collection *(Continued)*

Profile	OS	CPU	Memory	EPS	Endpoints polled	Average CPU	Memory
High EPS / Medium Device Count	Server 2019	8 cores	16 GB	5000	250	4.8%	36 MB
High EPS / High Device Count	Server 2019	8 cores	16 GB	5000	500	7.1%	60 MB

Software requirements

The following table describes the software requirements:

Table 10: Software requirements

Requirement	Description
Operating system	Windows Server 2022 (including Core) Windows Server 2019 (including Core) Windows Server 2016 (including Core) Windows Server 2012 (including Core) Windows 10
Distribution	One WinCollect agent for each Windows host.
Required user role permissions for installation	Administrator, or local administrator

NOTE: 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, Juniper 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 JSA Collecting Windows Events (WMI/ALE/WinCollect) forum. For more information, see https://support.microsoft.com/en-us/lifecycle/search.

Upgrading a WinCollect 7 agent to WinCollect 10

IN THIS SECTION

You can upgrade stand-alone WinCollect agents to WinCollect 10 from version 7.3.0 or later.

To upgrade to WinCollect 10, you must run the **.msi** upgrade script from an administrative command prompt.

NOTE: When you upgrade your WinCollect 7.3.x agent to WinCollect 10, the installer converts your **AgentConfig.xml** file to the new WinCollect 10 **AgentConfig.xml** format. The installer also makes a backup of the 7.3.x **AgentConfig.xml** and places it in a patch directory under the default installation location with the date and time of the upgrade.

Use the following shortcut to open the administrative command prompt:

1. Press Windows+R.
2. Type **cmd**.
3. Press Ctrl+Shift+Enter.
   With the **Administrator: Command Prompt** window open, use one of the following methods to run the WinCollect 10 **.msi** upgrade script.

Upgrading with the WinCollect 10 upgrade wizard

You can upgrade stand-alone WinCollect agents to WinCollect 10 from version 7.3.0 or later.

To upgrade to WinCollect 10, you must run the **.msi** upgrade script from an administrative command prompt.

NOTE: When you upgrade your WinCollect 7.3.x agent to WinCollect 10, the installer converts your **AgentConfig.xml** file to the new WinCollect 10 **AgentConfig.xml** format. The installer also makes a backup of the 7.3.x **AgentConfig.xml** and places it in a patch directory under the default installation location with the date and time of the upgrade.

To run the WinCollect 10 upgrade wizard, type the following command in the administrative command prompt:

```
msiexec.exe /i WinCollect-10.X.X-X.x64.msi
```

The upgrade wizard steps through the upgrade process. After the wizard completes, you can close the command prompt window.

Running the Silent Upgrade

You can upgrade stand-alone WinCollect agents to WinCollect 10 from version 7.3.0 or later.

To upgrade to WinCollect 10, you must run the **.msi** upgrade script from an administrative command prompt.

NOTE: When you upgrade your WinCollect 7.3.x agent to WinCollect 10, the installer converts your **AgentConfig.xml** file to the new WinCollect 10 **AgentConfig.xml** format. The installer also makes a backup of the 7.3.x **AgentConfig.xml** and places it in a patch directory under the default installation location with the date and time of the upgrade.

To run the WinCollect 10 silent upgrade, type the following command in the administrative command prompt:

```
msiexec.exe /qn /i WinCollect-10.X.X-X.x64.msi
```

The WinCollect agent is silently updated in the background. After the upgrade is complete, you can close the command prompt window.

Upgrading WinCollect 10 agents

You can upgrade existing WinCollect 10 stand-alone agents to the latest version by either running the interactive MSI installer or by using the command line.

To upgrade WinCollect 10 agents, you must open the Command Prompt as an administrator.

1. Copy the installation **.msi** file to the agent computer.
2. To open the command prompt as an administrator, right-click the **Start** charm, and then click **Command Prompt (Admin)**.
3. Upgrade the agent by performing one of the following actions:
   - To upgrade using the interactive MSI installer, type the following command:

     ```
     WinCollect-10.<x.x>-x.x64.msi
     ```

   - To upgrade using the command line, type the following command:

     ```
     msiexec.exe /qn /i wincollect-10.X.X-X.x64.msi
     ```

Installing WinCollect 10 using the GUI Quick installation

You can easily get WinCollect 10 up and running and sending events to JSA by selecting **Quick** during the installation and specifying the JSA appliance you want to send your events to.

1. Download the latest version of the WinCollect 10 agent.
2. Double-click the installer to begin the installation.
3. In the **Welcome to the IBM 64-bit Setup Wizard** window, click **Next**.
4. Accept the EULA and click **Next**.
5. In the **Installer Options** window, select **Quick**.

   The Quick installation option configures the agent to collect Security, System, and Application events.
6. In the **Destination Input** window, type in the **Hostname** of the JSA appliance you want to send your WinCollect events to, and click **Install**.
7. After the installation is complete, the **Completing the WinCollect Setup Wizard** window appears. Click **Finish** to close the installer.

Installing WinCollect 10 using the command line

You can quickly start collecting events from the Application, System, and Security event channels by using the command line to install WinCollect 10 using the command line.

1. Copy the installation **.exe** file to your computer.
2. Run the following command:

```
msiexec.exe /qn /i WinCollect-10.X.X-X.x64.msi QUICK_INSTALL="yes" WC_DEST="<qrhostname.domain.lab>"
```

> **NOTE**: You must change `WC_DEST=` to point to the JSA appliance where you want to send the events.

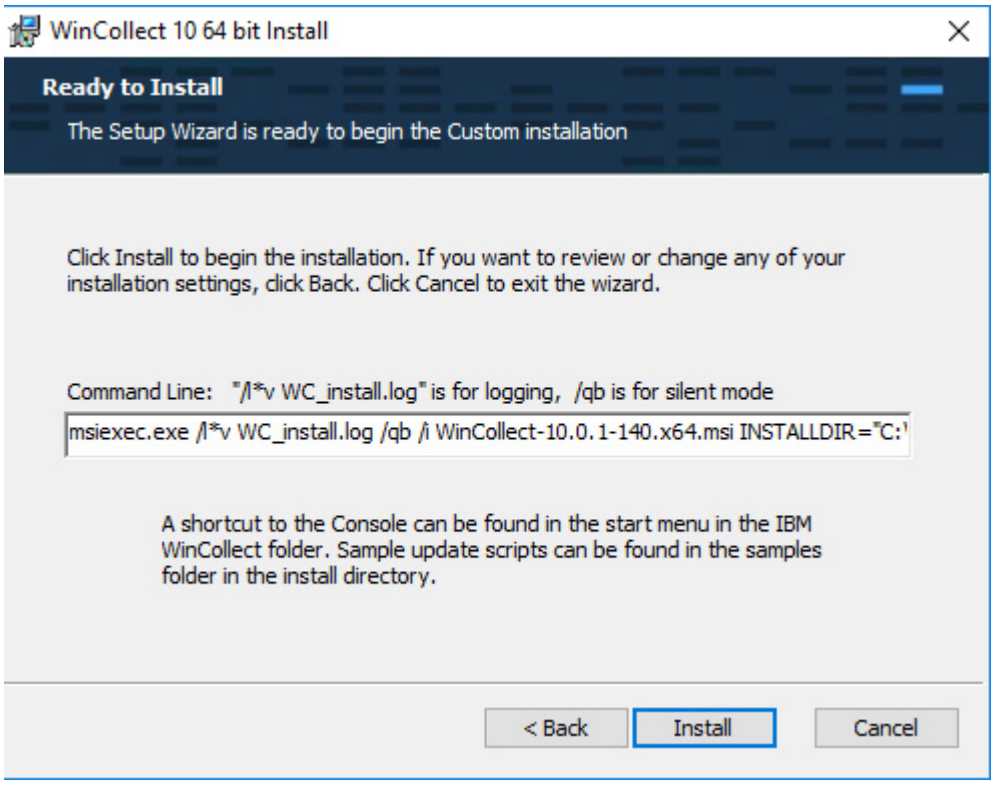
Installing WinCollect 10 using the Advanced installer

You can customize your WinCollect installation by selecting the Advanced option on the WinCollect graphical installer. You can select which WinCollect 10 components to install and where they are installed.

1. Download the latest version of the WinCollect 10 installer.
2. Double-click the installer to begin.
3. Click **Next**.
4. Accept the License Agreement and click **Next**.
5. In the **Installer Options** window, select **Advanced**.
6. In the **Custom Setup** window, select the components of the agent that you want to install.
7. Click **Next** to open the **Configuration Options** window, and then specify your startup agent configuration. Select one of the following options:
   - **Specify a configuration script file to execute immediately after the Agent is installed.** Use this option to upload a script file with a set of instructions for the agent that tell the what to collect or where to send the events etc. For more information, see "Installing or updating an agent using an update script" on page 49.
   - **Specify a destination to send Application, System, and Security events.** This option captures the Application, System and Security events for the local system and specifies which JSA application you would like to send them to. This is similar to using the Quick Installation option.
   - **Install the agent without a configuration.** This option installs the agent with a blank slate. Then, you must either use the Console to configure the agent or use an update script to configure the agent after installation.

8. Click **Next**.

> **TIP**: In the **Ready to Install** window, you can copy the CLI command that includes all the settings you chose during the Advanced Installation. Use that command to install other agents with the same configuration options that you specified in the Advanced Installer.



9. Click **Install** to complete the installation.

WinCollect 10 Command line installation advanced examples

IN THIS SECTION

Use one of the following examples to run a silent installation or specify an installation script that gives the agent instructions on what to monitor and where to send events.

Command line options

Parameter	Description
QUICK_INSTALL	
WC_DEST	Used to define the destination. Creates a default destination called JSA.
INSTALLDIR	Used to specify the installation directory.
WC_SCRIPT	Used to specify the update script file to be used for advanced installation.

Examples

Specify the Installation Directory

The following command runs a silent quick installation that collects Application, System, and Security events in the **C:\WinCollect** directory and specifies `qradarappliance.yourdomain.lab` as the destination. `msiexec.exe /qn /i wincollect-10.X.X-X.x64.msi INSTALLDIR="C:\WinCollect\" QUICK_INSTALL="yes" WC_DEST="qradarappliance.yourdomain.lab"`

Specify a Configuration Script

The following command runs a silent installation and specifies an installation script that gives the agent instructions on what to monitor and where to send events. `msiexec.exe /qn /i wincollect-10.X.X-X.x64.msi WC_SCRIPT="c:\Users\<youruseraccount>\Desktop\update_AddMSEvents_EnableDestination.xml"`

WinCollect 10 installation script examples

You can quickly install and configure WinCollect 10 by specifying an installation script when you install WinCollect by using the command line.

For example, you can use an installation script that specifies the following parameters to be configured during installation:

- Collect only the standard event logs.

- Collect standard event logs and sysmon.
- Collect standard event logs and DHCP.

You can download sample installation scripts and templates to use with most compatible plug-ins in the WinCollect 10 folder here: https://github.com/ibm-security-intelligence/wincollect/tree/master/.

For more information about using installation scripts during command-line installation, see "WinCollect 10 Command line installation advanced examples" on page 19.

CHAPTER 3

Uninstalling WinCollect 10

Uninstalling WinCollect 10

IN THIS SECTION

You can uninstall WinCollect 10 by using the command line, the Control Panel, or the Windows Start Menu.

Uninstalling WinCollect 10 using the command line

You can uninstall WinCollect 10 using the command line.

1. Open the **Command Prompt** with administrative rights.
2. Run the following command: `wmic product where "name='IBM WinCollect 10' " uninstall`

Uninstalling WinCollect 10 using the Control Panel

You can use the Control Panel to uninstall WinCollect 10.

1. Open the Control Panel and select **Programs** > **Uninstall a program**.
2. Select **IBM**.
3. Right-click and select **Uninstall**.

Uninstalling WinCollect 10 using the Start menu

You can uninstall WinCollect 10 using the link in the **Start** menu.

1. Click the **Start** menu and go to **IBM WinCollect 10**.

2. Right-click and select **Uninstall IBM WinCollect 10**.

4

CHAPTER

WinCollect 10 Stand-alone Console

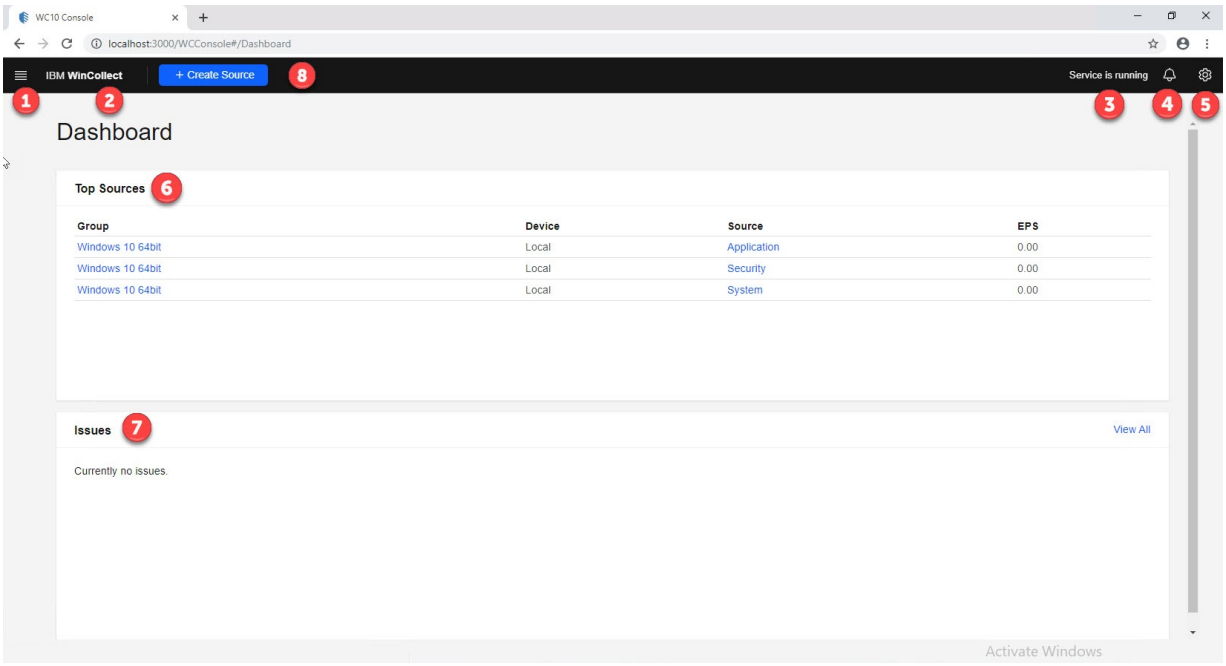
WinCollect 10 stand-alone console

IN THIS SECTION

The WinCollect 10 stand-alone console is automatically installed when you install WinCollect 10.

After installation, a **WinCollect 10** icon appears in the Windows **Start** menu.

Figure 2: The WinCollect 10 dashboard



The **Settings** menu (1) contains the following options:

- Agent configuration, including "Agent core" on page 31, "Security" on page 32, Local Sources, Remote Sources, Destinations, and advanced System Settings.
- "Log Viewer" on page 41
- Source wizard

Click **IBM WinCollect** (2) to return to the dashboard at any time.

The "Service Status" on page 40 message (3) provides a visual representation of the status of the WinCollect 10 service.

Click the **Notifications** icon (4) to see changes that need to be deployed to your agent.

Use the "Agent Settings" on page 38 menu (5) to configure log settings, generate files for Juniper Customer Support, toggle the Advanced user interface, and see the current build version that you are using.

The "Top Sources" on page 41 pane (6) displays the top five log sources, based on highest EPS.

The Issues pane (7) displays the latest warnings and errors that are detected by the agent.

Click **Create Source** (8) to launch the source wizard.

Opening the WinCollect 10 stand-alone console

IN THIS SECTION

After installation, a new WinCollect 10 folder appears in the Windows **Start** menu.

WinCollect 10 is designed to work on Microsoft Internet Explorer 11, Edge, Mozilla Firefox, and Google Chrome.

1. Click **Start**.
   The WinCollect 10 folder appears in the **Start** menu:
2. Click WinCollect **10 Console** to open the console.
   Use the console to configure and administer the agent.

> **TIP**: By default, the web service runs on port 3000. You can change the default port by editing the agent configuration options. You can also use the update script that is provided here to make this change.

Editing a source

Use the WinCollect 10 console to edit sources.

1. In the WinCollect 10 Console, click the menu icon and select **Local Sources**.

> **TIP**: To edit remote sources, select **Remote Sources** instead.

2. On the **Local Sources** window, select **Local source**.
   This source was created by default during the WinCollect installation.

The **Edit** view opens.

3. Select the source you want to edit, make the necessary changes, and click **Save**.
4. To deploy your changes, click the **notifications** icon, then click **Apply Changes**.

Adding a destination

If you installed the WinCollect 10 agent by using the "Quick Installation," a destination called JSA was created during the installation. You can also add a new destination.

1. From the **Agent Configuration** menu, select **Destinations**.
2. On the **Destinations** window, click **Add** in the **Destinations** section.
3. Enter the following required information:

Option	Description
Name	Give your destination a name.
Device Address	Use the DNS name of your JSA appliance. If DNS is not configured, you can use the IP address.
Protocol	The default is set to TCP. **NOTE**: If you want to specify a secondary destination, you must select the TCP Protocol.

For more information about other optional parameters, see the "Destination parameters" table.

4. Click **Save**.
5. Deploy your changes.

Table 11: Destination parameters

Name	Name of the destination.
Enabled	Enable or disable the destination.

Device Address	The hostname or IP address of the appliance where the event data is sent.
Port	The port number used to send data.
Secondary device address	Used as a backup destination. For more information, see "Adding a Secondary Destination" on page 37.
Maximum events per second	Controls the EPS rate sent to that destination.
Format	The format the destination uses to send data. For more information, see Destination Formats.
Include Agent ID	If configured, you can include the Agent Identifier in the Syslog header for all your log sources. For more information, see Include Agent ID.
Protocol	The protocol used by the destination.

WinCollect 10 stand-alone configuration

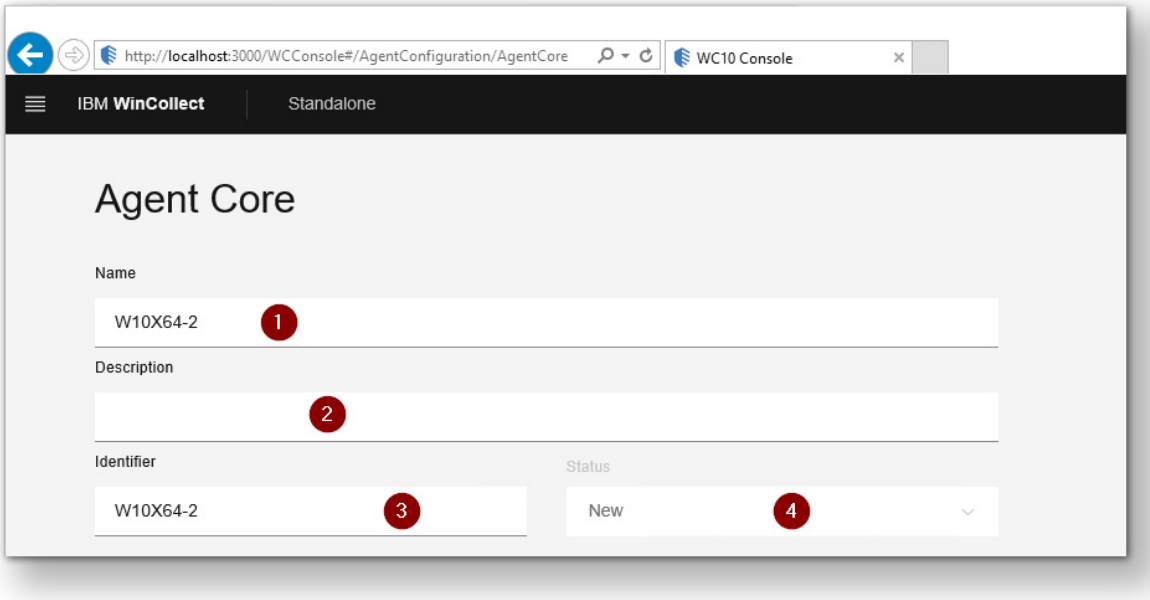
IN THIS SECTION

After you install a WinCollect 10 agent, you can configure the agent in the console or by using an update script.

Agent Core

IN THIS SECTION

The WinCollect 10 Agent Core is divided into several sections. Use the first section to configure specific agent settings.



The **Name** of the agent (1) is typically the same as the **Identifier**.

You can add a **Description** (2) of the agent.

The **Identifier** (3) is pulled automatically when you install the agent. This is usually the hostname value from the environment settings.

The **Status** list (4) is designed for use in future releases of WinCollect in managed mode.

Enabling statistics messages

WinCollect 10 can send agent statistics to JSA in the form of status messages.

1. Click **Settings** > **Agent Configuration** > **Agent Core**.
2. In the **Status Message** section, select **Statistics**.

3. Click **Save**.
4. Deploy your changes.
   WinCollect 10 starts sending status messages to your status server.

   ```
   <13>Jan 29 13:06:00 WCDEMODC LEEF:1.0|IBM|WinCollect|10.0.0.106|2|src=WCDEMODC os=Windows Server 2016 (Build 17763 64-bit) dst=wc1.canlab.ibm.com sev=3 log=System.WinCollect.Statistics msg=Target.QRadar=0 UserData.DiskSpaceUsed=0
   ```

   After you enable some sources, the status messages show events that are collected from multiple sources. In this example, `msg=FromSources=3` indicates that events are collected from three sources:

   ```
   <13>Jan 29 13:22:00 WCDEMODC LEEF:1.0|IBM|WinCollect|10.0.0.106|2|src=WCDEMODC os=Windows Server 2016 (Build 17763 64-bit) dst=wc1.canlab.ibm.com sev=3 log=System.WinCollect.Statistics Target.QRadar=3 UserData.DiskSpaceUsed=0
   ```

Credentials

IN THIS SECTION

Use the **Credentials** section of the console to manage the user accounts you need for remote polling the Windows systems.

TIP: Create a dedicated user account that is configured with the correct permissions to read the event logs and file permissions for the various logs you can collect with WinCollect.

Adding a user account

Use the **Credentials** section of the console to add the user accounts that you need.

1. Click **Add**, and type a **Name** for the account.

   TIP: The name is not used for authentication. You can use any name that you want.

2. If your account is part of a domain, specify the **Domain** name.
3. Type a **User name** and **Password**.
   After you save, a notification appears that you have pending changes.
4. Click the **Notifications** icon and select **Apply Changes** to deploy the changes to the agent.

Configuring a local source

If you didn't use the Quick Installation option to install your agent, you can still configure the agent to collect System, Application, and Security events on the local asset. You can use the Configuration Console or run an update script to configure the agent.

Use the following procedure to configure the agent by using the Configuration Console. To configure an agent by using an update script, see "Agent configuration update script use cases" on page 49.

1. On the WinCollect 10 Console, click the menu icon and select **Local Sources**.
2. In the **Local Sources** window, select **Local source**, which was created by default during the WinCollect installation.
3. In the **Edit** view, click **Add** to configure the event channels that you want to collect.
   a. In the **Source Collection** window, select **Channel** > **Application**, and click **Save**.

      TIP: You do not need to enter a **Name** or **Identifier**.

   b. Select **Channel** > **Security**, and click **Save**.
   c. Select **Channel** > **System**, and click **Save**.

      TIP: In WinCollect 10, you can configure each channel (source) differently. In previous WinCollect versions, all channels used the same configuration set.

   The three channels are now listed in the **Source** section.
4. You must configure the **Destination**, using information from your JSA deployment:
   You can specify any application in your deployment where you like to send the event to.
   a. Click the **menu** icon, and select **Destinations**.
   b. Select the default **JSA** destination.
   c. On the **Edit** window, select **Enabled**.
   d. Type the **Hostname** or **IP address** of the JSA appliance that you want to send the events to.

      TIP: Leave all other options on their default values.

   e. Click **Save**.
   f. On the **Destinations** window, click **Save**.

5. To deploy your changes, click the **notifications** icon, then click **Apply Changes**.

RELATED DOCUMENTATION

Installing WinCollect 10 using the GUI Quick installation | 17

Configuring a remote source

You can configure sources to remotely collect Windows events in the WinCollect 10 Console.

Ensure that the user account that you are using has permissions to connect to the remote devices that are configured in Step 10.

1. From the WinCollect 10 Console, click the menu icon, and select **Source Wizard**.
2. Select **Remote** for the **Select Source Group Type**.
3. For **Select Source Group**, click **Create New**.

   TIP: You can also add the new device to an existing group.

4. Type **Domain Workstations** as the name of the group, and add a description.
5. On the **Select Source Type** window, leave the default settings to **Windows Event Subscription**.
6. In the **Configure Source Parameters** section, select the channels that you want to collect events from.

   TIP: You can also create an XPath Query that contains a custom set of channels and event IDs that you want to create.

7. Select the **Application**, **System**, and **Security** event channels, then click **Credentials**.
8. Click **Create New**.

   TIP: If you previously added a credential, select it in the **Select Credentials** window. By default, after you install a new agent, no credentials are configured.

9. In the **Credentials** window, enter the credential details and click Step 6: **Device**.
10. In the **Create Device** window, enter the following details for device that you want to collect events from:

Option	Description
Device Address	Specify the FQDN or the IP address of the remote device.

Option	Description
Name	If you don't specify a name, the FQDN or IP address from the **Device Address** is used as the name.
Description	(Optional) Type a description to identify the device.
Credentials	(Optional) Specify the credentials that you created in the previous step.

11. In the **Configure Destination** window, specify where you want these events to go.

    TIP: If you installed the agent using the Quick Installation, you might already have a **Destination** created called JSA. If you want your new remote source to go to the same location, you can select this destination.

    a. To add another JSA appliance, select **Create New**.

    b. Type **QRadarEP** as the **Name**.

    c. Add a **Description**.

    d. Specify the hostname or the IP address of the JSA appliance as the **Device Address**.

    TIP: If you are using the hostname of the EP, ensure that your agent can resolve the hostname. The default port number is 514. The default Maximum events per second is 20,000.

12. Click **Finish**.
    The WinCollect 10 dashboard displays a notification that you have pending changes.
13. Deploy the changes.

RELATED DOCUMENTATION

Configuring a remote source with an update script | 59

Destinations

IN THIS SECTION

Destinations are any JSA appliance in your deployment where you want to send your event data. You can send syslog event data using UDP, TCP, or TLS protocols.

Adding a destination

If you installed the WinCollect 10 agent by using the "Quick Installation," a destination called JSA was created during the installation. You can also add a new destination.

1. From the **Agent Configuration** menu, select **Destinations**.
2. On the **Destinations** window, click **Add** in the **Destinations** section.
3. Enter the following required information:

Option	Description
Name	Give your destination a name.
Device Address	Use the DNS name of your JSA appliance. If DNS is not configured, you can use the IP address.
Protocol	The default is set to TCP. **NOTE**: If you want to specify a secondary destination, you must select the TCP Protocol.

   For more information about other optional parameters, see the "Destination parameters" table.
4. Click **Save**.
5. Deploy your changes.

Table 12: Destination parameters

Name	Name of the destination.
Enabled	Enable or disable the destination.

Device Address	The hostname or IP address of the appliance where the event data is sent.
Port	The port number used to send data.
Secondary device address	Used as a backup destination. For more information, see "Adding a Secondary Destination" on page 37.
Maximum events per second	Controls the EPS rate sent to that destination.
Format	The format the destination uses to send data. For more information, see Destination Formats.
Include Agent ID	If configured, you can include the Agent Identifier in the Syslog header for all your log sources. For more information, see Include Agent ID.
Protocol	The protocol used by the destination.

Deleting a destination

If you are no longer using an JSA destination, you can delete it from the **Destinations** list.

1. From the **Agent Configuration** menu, select **Destinations**.
2. Select the destination that you want to delete, and click **Delete**.
3. Deploy your changes.

> **NOTE**: If you try to delete a destination that is being used by a source, an error message appears in the **Pending Changes** section.

Adding a secondary destination

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

1. In the WinCollect 10 Console, click the menu icon and select **Destinations**.
   The default installation of WinCollect 10 contains a destination that is named JSA. You can use this destination or add a new destination.
2. Select the default JSA destination to open the **Properties** window.
3. On the **Destination Properties** window, configure the following options:

Option	Description
Secondary Device Address	Type the hostname or IP address of the secondary destination. The destination can be an appliance in the same deployment or a separate deployment. If the agent can connect to the IP address or hostname, the agent fails over to this appliance if it loses connection with the primary Device Address.
Failover timeout (seconds)	The amount of time the agent waits before failing over to the **Secondary Device Address**. The default is 1800 seconds (30 minutes). After the agent fails over to the secondary device, it checks every 5 minutes for a connection to the primary device.

NOTE: For a secondary destination, you must select the TCP Protocol.

4. Click **Save**.
5. Deploy your changes.

Send events to multiple destinations

If you want WinCollect 10 to send events simultaneously to two separate JSA deployments, you must duplicate your source group.

To duplicate your source group, use the "Source wizard" on page 43 to create the new local or remote group, then select a different destination to send the events.

NOTE: The agent uses different polling interval times for each of the duplicated sources. Also, the new group might assign a different identifier to the agent than the original group.

Agent settings

IN THIS SECTION

Use the **Agent Settings** menu to configure the log settings for the agent, turn on the **Advanced UI**, or view the version that is installed.

Collecting files to send to Juniper Customer Support

Use the Collect Support Files option to quickly capture the WinCollect logs and send them to Juniper Customer Support to help troubleshoot any issues you experience.

1. Click the **Settings** icon, then click **Collect Support Files**.
2. Click **Collect and compress files** to create a compressed (.zip) file in the logs directory, inside the default installation directory.
   When the operation is successfully completed, a notification with the path and the name of the file is displayed.

Configuring logs

Use **Log Configuration** to quickly change the log levels for the three main categories: Code, Device, and Console.

Adjust log settings only when instructed by a Juniper Customer Support representative.

1. Click the **Settings** icon, then click **Log Configuration**.
2. Select a value in the **Set all Categories to** list to quickly change the three main categories.
3. If you want to change individual categories, expand **Advanced log configuration**.
4. To add specific categories, select them in the menu and click **Add Category**.

> **TIP**: Use this option to troubleshoot the agent with assistance from a Juniper Customer Support representative.

5. To include the log in a WinCollect status message that is sent to JSA, select **Feed as Status Message**.
6. Click **Save**.

After you save the log configuration settings, you do not need to restart or deploy any changes to the agent.

Advanced UI

To simplify the WinCollect 10 user interface, many finer control settings are included in the advanced UI.

Advanced users who want finer control of their WinCollect 10 agents can turn on the advanced UI by clicking the **Settings** icon, then moving the **Advanced UI** switch to the **On** position.

NOTE: Use the **Advanced UI** options only under direction from Juniper Customer Support.

Service status

In the Service Status section of the console, you can quickly see the status of the WinCollect agent service.

If you select a service that is running, you can **Stop** or **Restart** the service. The service status now shows **Service is stopped**.

If you select a service that is stopped, you can click **Start**. The service starts and a new **Startup** window is displayed.

Table 13: Included services

Service	Description
Delete Logs	Displays the number of **WinCollect** and **WCConsole** log files in the log directory. You can delete the logs before the agent starts by selecting **Delete Logs**. Use this service to troubleshoot the agent and making sure you have a clean set of log files.
Delete Patch/Staging	This service is not applicable for stand-alone agents.
Deleted Cached Events	Displays the total number of events that were not sent to their destination. To delete the cached events, select **Delete Cached Events**. **NOTE**: If files are listed here, those events are deleted. You might be able to collect those events again if you select **Delete Bookmarks** and the log file has not rolled over.
Delete PEM File	This service is not applicable for stand-alone agents.

Table 13: Included services *(Continued)*

Service	Description
Delete Bookmarks	Displays the total number of bookmark files for the sources the agent is responsible for collecting events from. Use this service to delete the bookmarks before the agent starts. **NOTE**: Deleting bookmarks might cause the agent to reread some of the files again and send events that were already sent to your destination.
Start in Debug mode	Puts the agent in Debug mode when it restarts.

Log Viewer

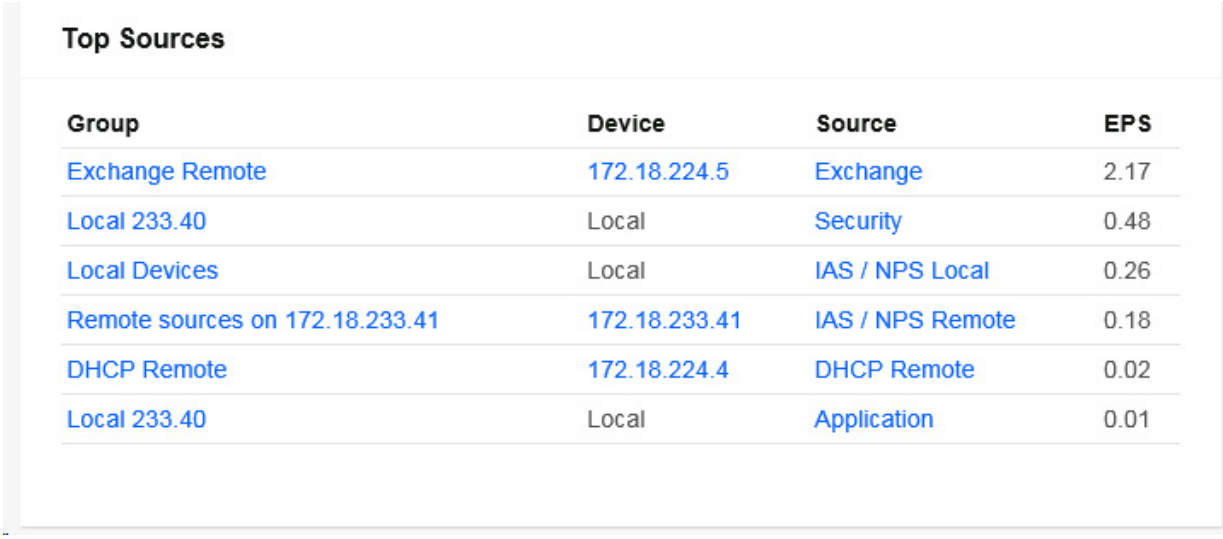
Use the Log Viewer to view the WinCollect.log file without the need for a text editor.

To open the **Log Viewer**, click the menu icon, then select **Log Viewer**.

Top Sources

The Top Sources list displays a maximum of 10 sources with the highest number of events per second (EPS).

You can click on the source name to directly open the source configuration, or click the Device or Group to open their configuration pages.



Top Sources

Group	Device	Source	EPS
Exchange Remote	172.18.224.5	Exchange	2.17
Local 233.40	Local	Security	0.48
Local Devices	Local	IAS / NPS Local	0.26
Remote sources on 172.18.233.41	172.18.233.41	IAS / NPS Remote	0.18
DHCP Remote	172.18.224.4	DHCP Remote	0.02
Local 233.40	Local	Application	0.01

Applying pending changes

After you make certain changes in the WinCollect 10 Console, you must apply the pending changes before they take effect.

1. Click the **Notifications** icon to go to the **Pending changes** window, where you can review the changes that are waiting to be deployed to your agent.
2. If you have pending changes, click **Apply Changes**.

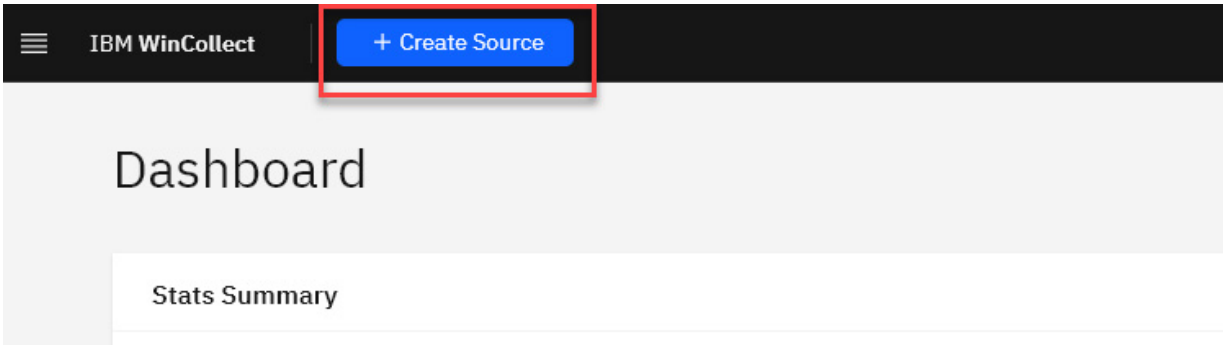
> **TIP**: If you decide you no longer need to apply changes, click **Discard Changes**. The window then displays `No pending changes`. Click **Close**.

After the changes are applied, a new banner displays a confirmation that the changes were successfully applied. The agent creates a patch folder with the new changes to be processed, which are shown in the **Success** section of the banner.

Create a source in the Source wizard

IN THIS SECTION

You can use the WinCollect 10 Source wizard to quickly create a source by clicking **Create Source**.



The wizard guides you through creating either a local source or one or more remote sources. Create a **Local** source if the Agent is installed on the local machine. To collect data from a different machine, create a **Remote** source.

Creating a local source

Use the WinCollect 10 Source wizard to create a **Local** source if the agent is installed on the local device.

1. Click **Create Source**.
2. Select or create a **Source Group** to keep your related sources together.

   You can edit sources in bulk that are in the same group.
3. Select a **Source Type**.

   The **Source Type** defines the type of logs you want to collect. For example, if you want to collect standard Windows events then select **Microsoft Windows Events**.
4. Configure the **Source Parameters**.

Microsoft Windows Events	Monitor common event channels within the Windows event logging system, including XML queries (XPath).
5. Enable the channels that you want to collect from.
6. Select a **Destination**.

WinCollect destinations define the parameters for how the WinCollect agent forwards events and logs to an JSA appliance.

> **TIP**: If you select an existing destination that is disabled, the wizard enables it while configuring the source.

7. Click **Summary** to view a list of the changes you made.
8. Click **Apply**.

Creating a remote source

To collect events from a different device that doesn't have the WinCollect 10 agent installed, create a **Remote** source.

When you set up a **Remote** source, you have the option to add multiple remote sources at the same time.

> **NOTE**: The following steps outline the procedure to add multiple sources (**Bulk Add**). Adding a single device is very similar.

1. Click **Create Source**.
2. Select the **Remote Type:**

Option	Description
Single	Add one remote source.
Bulk	Add up to 500 remote sources.

3. Select or create a **Source Group** to keep your related sources together.

   You can edit sources in bulk that are in the same group.
4. Select a **Source Type**.

   The **Source Type** defines the type of logs you want to collect. For example, if you want to collect standard Windows events then select **Microsoft Windows Events**.
5. Configure the **Source Parameters**.

Microsoft Windows Events	Monitor common event channels within the Windows event logging system, including XML queries (XPath).

6. Enable the channels that you want to collect from.
7. Select **Credentials**.

   A credential contains login information that the WinCollect agent uses to connect to remote devices.

8. Select **Bulk Add Devices**.

   You can either upload a text file with a list of devices to add one per line, or manually enter them one at a time and click **Add**. Bulk Template file example (**WinCollectBulkAddSample.txt**)

```
172.18.100.200
172.18.100.201
test.example.net.workstation1
172.18.100.202
172.16.200.100
172.16.200.101
test.example.net.workstation2
172.18.200.102
```

9. To perform a connection test, select the devices that you want to test.

   The test is performed when you proceed to the next step.

   NOTE: A failed connection can take up to 20 seconds to time out.

10. Select a **Destination**.

    WinCollect destinations define the parameters for how the WinCollect agent forwards events and logs to an JSA appliance.

    TIP: If you select an existing destination that is disabled, the wizard enables it while configuring the source.

11. Click **Summary** to view a list of the changes you made.
12. Click **Apply**.

5
CHAPTER

Configuration Scripts

Configuration scripts

IN THIS SECTION

You can change WinCollect 10 agent configuration without making manual or scripted edits to the AgentConfig.xml file by using update scripts.

When you copy an update script to the WinCollect patch directory, the agent completes the actions described in your script.

Configuring WinCollect 10 to collect Microsoft security events

If you use the "Installing WinCollect 10 using the Advanced installer" on page 18 option to install your agent, you can run an update script to configure the agent.

1. Download or copy the **wincollect-10.0.*x.x*64.exe** file to your computer.
2. Copy the update script code that is displayed in this topic and paste it into a text editor.
3. Replace the value for the `Address` parameter (`"YourStatusServerIP"`) with the IP address of an appliance in your JSA deployment.
4. Replace the `Destination Address` parameter (`"YourQRadarApplianceIP"`) with the IP address of an appliance in your JSA deployment.

   NOTE: If you are using an All-In-One appliance, the `Destination Address` can be the same IP address as the `Address` parameter.

5. Save the file as **update_localmsevents.xml**.
6. Run the **wincollect-10.0.*x.x*64.exe** installer as an admin user.
7. On the **Welcome to the WinCollect 10 Setup Wizard** window, click **Next** and accept the terms in the license agreement.
8. Enter your **Company Information**, then click **Next**.

9. On the **Custom Setup** window, specify an alternative path to install and choose any additional components you need to install.
10. Click **Next**.
11. On the **Configuration Options** window, select **Specify a configuration script file to execute immediately after the Agent is installed**.
12. Click **Browse** to locate your **update_localmsevents.xml** file and click **Next**.

> **TIP**: The **Ready to Install** window displays the command that you can use to apply the same configuration on another agent.

13. Click **Install** to finish the installation.
14. If you are prompted to allow the app to install from an unknown publisher, click **Yes**.
15. On the **Completing the WinCollect 10 Setup Wizard** page, click **Finish**.

Copy the following code and save the file as **update_localmsevents.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
<Update objPath="AgentCore/StatusServer" >
<Parameter name="Address" value="YourStatusServer" />
<Parameter name="Enabled" value="true" />
</Update>
<Update objPath="Destinations" >
<Destination Name="QRadar" Address="YourDestinationServer" Enabled="true" />
</Update>
<Update objPath="ROOT" >
<LocalSources Name="Local" Type="MSEVEN6" Enabled="true">
<Source Channel="Application" />
<Source Channel="System" />
<Source Channel="Security" />
<Target Destination="QRadar" />
</LocalSources>
</Update>
</WinCollectScript>
```

Agent configuration update script use cases

IN THIS SECTION

You can change the agent configuration without making manual or scripted edits to the **AgentConfig.xml** file by using update scripts.

When you copy an update script to the WinCollect patch directory, the Agent performs the actions that are described in your script.

Update script actions

Update scripts can include the following actions:

<Addto> - Adds a child to the relevant node. You can include a type and certain defined key value pairs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
</WinCollectScript>
```

<Update> - Updates the value of a key.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
</WinCollectScript>
```

<Delete> - Similar to **<Addto>**, except you add the name of the object to the object Path.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
</WinCollectScript>
```

The agent creates a new configuration in the patch folder and validates the new configuration. When the validation is successful, the old agent configuration is backed up. The agent moves the update script, the old **AgentConfig.xml** file, and the new **AgentConfig.xml** file into a backup folder (**patch_*xxxx***) and puts the new agent configuration into operation.

Several sample update scripts are installed with WinCollect 10. They are stored in the **\IBM\WinCollect \samples** directory.

Examples

NOTE: These scripts are examples only. All agent configuration service modules are supported so that you can create your own scripts.

NOTE: Update script file names must begin with **update_**. All other file names are ignored.

update_addXPath.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
    <AddTo objPath="LocalSources(Name=Local)" >
        <Source Name="Security XPath" >
            <Parameter name="Query">
                <QueryList>
                  <Query Id="1" Path="Security">
                    <Select Path="Security">*[System[(Level=1  or Level=2 or Level=3) and
```

```
(EventID=1 or EventID=3 or  (EventID >= 5 and EventID <= 100)  or  (EventID >= 200 and EventID
<= 500) )]]</Select>
                  </Query>
                </QueryList>
            </Parameter>
        </Source>
    </AddTo>
</WinCollectScript>
```

update_delSecurity.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
    <Delete objPath="LocalSources(Name=Local)/Source(Name=Security)" />
</WinCollectScript>
```

update_updtConsPort.xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<WinCollectScript version="10.0">
    <Update objPath="SystemSettings" setParam="Console.Port" value="3333" descr="Changing the
Config Console's port to 3333" />
</WinCollectScript>
```

The following use cases are examples of how you can use update scripts to change agent configurations:

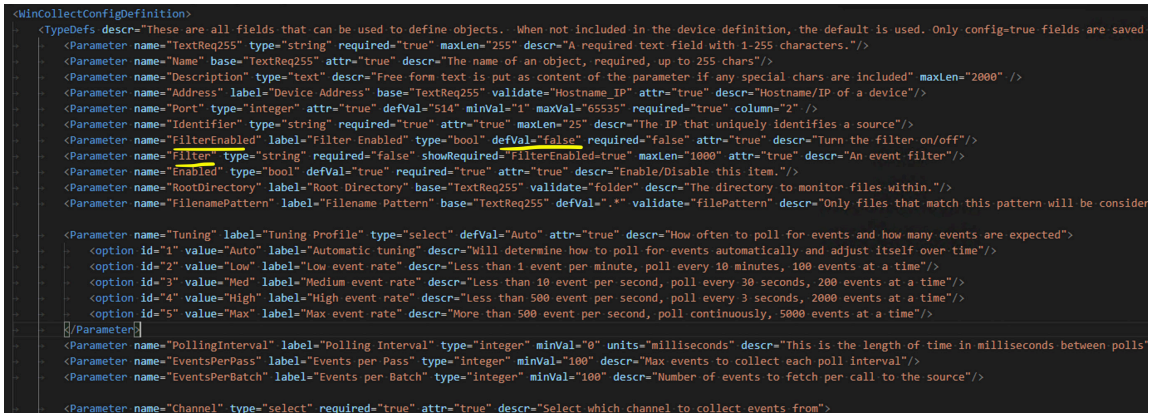
Adding NSA filtering to an existing source

You want to add NSA filtering to an existing source. You can change this attribute by using the **update_updtTemplate.xml** update script.

1. Locate the **update_updtTemplate.xml** template in the **\IBM\WinCollect\samples** directory.
2. Save a copy of the template and name it **update_addedNSAFilterToSecurity.xml**.
3. Modify the file:
   a. Open the agent config definition file (**AgentConfigDefinition.xml**) and find the parameter that you want to modify.

> **NOTE**: Do not modify the **AgentConfigDefinition.xml** file.

b. The `Filter` and `FilterEnabled` flags are in the `TypeDef` object, which means that every object can call upon the default value. This means that the `Source` object for security has these parameters, and the default values are as shown here. To refer to a child object, use a forward slash (/). The default value of the `FilterEnabled` parameter is `true`, so you need to change only the filter itself.



```
<WinCollectConfigDefinition>
    <TypeDefs descr="These are all fields that can be used to define objects.  When not included in the device definition, the default is used. Only config=true fields are saved
        <Parameter name="TextReq255" type="string" required="true" maxLen="255" descr="A required text field with 1-255 characters."/>
        <Parameter name="Name" base="TextReq255" attr="true" descr="The name of an object, required, up to 255 chars"/>
        <Parameter name="Description" type="text" descr="Free form text is put as content of the parameter if any special chars are included" maxLen="2000" />
        <Parameter name="Address" label="Device Address" base="TextReq255" validate="Hostname_IP" attr="true" descr="Hostname/IP of a device"/>
        <Parameter name="Port" type="integer" attr="true" defVal="514" minVal="1" maxVal="65535" required="true" column="2" />
        <Parameter name="Identifier" type="string" required="true" attr="true" maxLen="25" descr="The IP that uniquely identifies a source"/>
        <Parameter name="FilterEnabled" label="Filter Enabled" type="bool" defVal="false" required="false" attr="true" descr="Turn the filter on/off"/>
        <Parameter name="Filter" type="string" required="false" showRequired="FilterEnabled=true" maxLen="1000" attr="true" descr="An event filter"/>
        <Parameter name="Enabled" type="bool" defVal="true" required="true" attr="true" descr="Enable/Disable this item."/>
        <Parameter name="RootDirectory" label="Root Directory" base="TextReq255" validate="folder" descr="The directory to monitor files within."/>
        <Parameter name="FilenamePattern" label="Filename Pattern" base="TextReq255" defVal=".*" validate="filePattern" descr="Only files that match this pattern will be consider

        <Parameter name="Tuning" label="Tuning Profile" type="select" defVal="Auto" attr="true" descr="How often to poll for events and how many events are expected">
            <option id="1" value="Auto" label="Automatic tuning" descr="Will determine how to poll for events automatically and adjust itself over time"/>
            <option id="2" value="Low" label="Low event rate" descr="Less than 1 event per minute, poll every 10 minutes, 100 events at a time"/>
            <option id="3" value="Med" label="Medium event rate" descr="Less than 10 event per second, poll every 30 seconds, 200 events at a time"/>
            <option id="4" value="High" label="High event rate" descr="Less than 500 event per second, poll every 3 seconds, 2000 events at a time"/>
            <option id="5" value="Max" label="Max event rate" descr="More than 500 event per second, poll continuously, 5000 events at a time"/>
        </Parameter>
        <Parameter name="PollingInterval" label="Polling Interval" type="integer" minVal="0" units="milliseconds" descr="This is the length of time in milliseconds between polls"
        <Parameter name="EventsPerPass" label="Events per Pass" type="integer" minVal="100" descr="Max events to collect each poll interval"/>
        <Parameter name="EventsPerBatch" label="Events per Batch" type="integer" minVal="100" descr="Number of events to fetch per call to the source"/>

        <Parameter name="Channel" type="select" required="true" attr="true" descr="Select which channel to collect events from">
```

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectConfiguration version="10.0.0" xmlns="event_collection/WinCollect" >
    <AgentCore Name="WCJUMPNICK" Identifier="WCJUMPNICK" >
        <StatusServer Address="www.test.com" Heartbeat="60" Protocol="TCP" />
    </AgentCore>
    <Destinations >
        <Destination Name="QRadar" Address="www.test.com" />
    </Destinations>
    <LocalSources Name="Local" Type="MSEVEN6" >
        <Target Destination="QRadar" />
        <Source Channel="Application" />
        <Source Channel="System" />
        <Source Channel="Security" />
    </LocalSources>
    <Security >
        <Parameter name="Description">
            This will hold credentials for remote collection
        </Parameter>
    </Security>
    <SystemSettings >
        <Parameter name="Description">
            System setting overrides
        </Parameter>
    </SystemSettings>
</WinCollectConfiguration>
```

c. Change the object path to `LocalSources(Name="Local")/Source(Channel=Security)`.

d. Change the value of the `FilterEnabled` parameter to true, and the value of the `Filter` parameter to *<NSA_FILTER_SECURITY>*.

e. Change the description for the update to `Adding NSA filter to security channel on local sources.` The final script looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.0" >
    <Update objPath="LocalSources(Name=Local)/Source(Name=Security)" >
        <Parameter name="Filter"
value="1100,1102,4624,4625,4634,4648,4657,4672,4688,4689,4704,4706,4713,4714,4716,4719,4720
,4722,4725,4726,4728,4731,4732,4733,4735,4740,4756,4765,4766,4767,4769,4776,4778,4779,4781,
4782,4793,4870,4873,4874,4880,4881,4882,4885,4886,4887,4888,4890,4891,4892,4896,4897,4898,4
899,4900,5038,5136,5137,5138,5139,5140,5141,5142,5144,5145,5376,5377,5632,6272,6273,6274,62
```

```
75,6276,6277,6278,6279,6280,6281" />
        <Parameter name="FilterEnabled" value="true" />
    </Update>
</WinCollectScript>
```

4. Save the **update_addedNSAFilterToSecurity.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***).

   System Events NSA Filter

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.0" >
    <Update objPath="LocalSources(Name=Local)/Source(Name=System)" >
        <Parameter name="Filter"
value="1,6,12,13,19,104,219,1001,1125,1126,1129,7000,7022,7023,7024,7026,7031,7032,7034,704
5" />
        <Parameter name="FilterEnabled" value="true" />
    </Update>
</WinCollectScript>
```

   Application Events NSA Filter

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.0" >
    <Update objPath="LocalSources(Name=Local)/Source(Name=Application)" >
        <Parameter name="Filter"
value="1,2,865,866,867,868,882,1000,1001,1002,1022,1033,1511,1518" />
        <Parameter name="FilterEnabled" value="true" />
    </Update>
</WinCollectScript>
```

Add Sysmon to your existing Windows event sources

You can use an update script to configure agents to collect Sysmon events.

To collect Sysmon events along with your System, Application, and Security events, add the following update script to your patches directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.0" >
    <AddTo objPath="LocalSources(Name=Local)" >
        <Source Name="Sysmon" Channel="XPath" Type="MSEVEN6" >
            <Parameter name="Query">
                <QueryList>
                    <Query Id="0" Path="Microsoft-Windows-Sysmon/Operational">
                        <Select Path="Microsoft-Windows-Sysmon/Operational">*</Select>
                    </Query>
                </QueryList>
            </Parameter>
        </Source>
    </AddTo>
</WinCollectScript>
```

This script adds Sysmon to your **Local** sources.

Changing the heartbeat interval

You want to change the heartbeat interval from 5 minutes to 1 hour on all deployed systems. You can change this interval by using the **update_updtTemplate.xml** update script.

1. Locate the **update_updtTemplate.xml** template in the **\IBM\WinCollect\samples** directory.
2. Save a copy of the template and name it **update_heartbeat.xml**.
3. Modify the file:
   a. Open the agent config definition file (**AgentConfigDefinition.xml**) and find the `Heartbeat` parameter.

      NOTE: Do Not Modify the **AgentConfigDefinition.xml** file.

      The `Heartbeat` parameter is in the `StatusServer` object.

   b. Change the object path in your script to `StatusServer`, the parameter to `Heartbeat`, and the value to `3600` (1 hour in seconds).
      The final script looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.0" >
```

```
    <Update objPath="LocalSources(Name=Local)/Source(Name=Security)" >
        <Parameter name="Filter"
value="1100,1102,4624,4625,4634,4648,4657,4672,4688,4689,4704,4706,4713,4714,4716,4719,4720
,4722,4725,4726,4728,4731,4732,4733,4735,4740,4756,4765,4766,4767,4769,4776,4778,4779,4781,
4782,4793,4870,4873,4874,4880,4881,4882,4885,4886,4887,4888,4890,4891,4892,4896,4897,4898,4
899,4900,5038,5136,5137,5138,5139,5140,5141,5142,5144,5145,5376,5377,5632,6272,6273,6274,62
75,6276,6277,6278,6279,6280,6281" />
        <Parameter name="FilterEnabled" value="true" />
    </Update>
</WinCollectScript>
```

4. Save the **update_heartbeat.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***).

Modifying the event data storage configuration

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. You can make these changes by using the **update_updtTemplate.xml** update script.

1. Locate the **update_updtTemplate.xml** template in the **\IBM\WinCollect\samples** directory.
2. Save a copy of the template and name it **update_dataLocationAndCapacity.xml**.
3. Modify the file:
   a. Open the agent config definition file (**AgentConfigDefinition.xml**) and find the parameter that you want to modify.

      NOTE: Do Not Modify the **AgentConfigDefinition.xml** file.

      The UserData.Location and UserData.Events.Capacity flags are in the SystemSettings object.

> **NOTE**: `%ALLUSERSPROFILE%` is an environment variable. The default value is `C:\ProgramData`. In this use case, change this value to `C:/WinCollect/Data`.

b. In your script, change the value of the `UserData.Location` parameter to `C:\Wincollect\Data`, and the description to `Changed the stored wincollect data to C:/Wincollect/Data`.

c. Change the `UserData.Events.Capacity` parameter to `20480` (20GB in MB) and the description to `Increased data capacity to 20GB (20480MBs)`.
The final script looks like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<WinCollectScript version="10.0">
    <Update
        objPath="SystemSettings"
        setParam="UserData.Location"
        value="C:/WinCollect/Data"
        descr="Changed the stored wincollect data to C:/Wincollect/Data."/>
    <Update
        objPath="SystemSettings"
        setParam="UserData.Events.Capacity"
        value="20480"
        descr=Increased data capacity to 20GB (20480MBs)"/>
</WinCollectScript>
```

4. Save the **update_dataLocationAndCapacity.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***).

Sending Syslog data to JSA over TCP

You want to send Syslog data to your JSA Status Server over TCP, rather than UDP. You must also specify this option in the Destination Manager on your JSA Console.

By default, log sources are sent by TCP. Status server messages are sent by UDP.

1. Locate the **update_updtTemplate.xml** template in the **\IBM\WinCollect\samples** directory.
2. Save a copy of the template and name it **update_ChangeUDPtoTCP.xml**.
3. Modify the file:
   a. Open the agent config definition file (**AgentConfigDefinition.xml**) and find the Protocol parameter.

> **NOTE**: Do Not Modify the **AgentConfigDefinition.xml** file.

The Protocol parameter is in the TypeDef object which means that every object can call upon the default value. The StatusServer object in AgentCore has a Protocol parameter, with a default value of UDP. To refer to a child object, use a forward slash (/).

b. Change the object path in your script to `AgentCore/StatusServer`, the `Protocol` parameter to `TCP`, and the description to `Changing status server protocol to TCP`.
The final script looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0">
    <Update objPath="AgentCore/StatusServer" setParam="Protocol" value="TCP" />
</WinCollectScript>
```

4. Save the **update_ChangeUDPtoTCP.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***).

Change the console port number

Add the following XML update script to the patches directory to change the running console port number to 3333.

TIP: Verify that the port isn't already being used on the host computer.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<WinCollectScript version="10.0">
    <Update objPath="SystemSettings" setParam="Console.Port" value="3333" descr="Changing the
Config Console's port to 3333" />
</WinCollectScript>
```

Configuring a remote source with an update script

IN THIS SECTION

You can use an update script to add a remote source group, add a single remote source device, or add multiple remote source devices.

Add a remote source group

This example quickly creates a new remote source group called "Domain Workstations," specifies which channels to collect events from, specifies a credential, and sets a destination to send the events to.

NOTE: You must use the Console UI to first set your credentials. This is the only way that you can create your password hash. For more information, see "Adding a user account" on page 32.

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
    <AddTo objPath="Destinations" >
        <Destination Name="QRadarEP" Address="qradarep.yourdomain.lab" />
    </AddTo>
    <AddTo objPath="ROOT" >
        <RemoteSources Name="Domain Workstations" />
    </AddTo>
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Target Destination="QRadarEP" />
    </AddTo>
    <AddTo objPath="Security" >
        <Credentials Name="wcservice" Domain="yourdomain.lab"
Password="(ET1)bMfEpIr2JE20vlAqwDPatw==" UserID="wcservice" />
    </AddTo>
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Device Address="ws01.yourdomain.lab" Credentials="wcservice" />
    </AddTo>
```

```
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Source Channel="Application" Type="MSEVEN6" />
    </AddTo>
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Source Channel="System" Type="MSEVEN6" />
    </AddTo>
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Source Channel="Security" Type="MSEVEN6" />
    </AddTo>
</WinCollectScript>
```

Add a remote source device

This example update script adds a remote source device to collect events from, adds a workstation that is named **ws02.yourdomain.lab**, and adds it to the same remote source group as the previous script, , using the same **Domain Workstations** credentials, `wcservice`.

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Device Address="ws02.yourdomain.lab" Credentials="wcservice" />
    </AddTo>
</WinCollectScript>
```

The remote source group contains the values for the events to collect and the user account to use for the credentials.

Add multiple remote source devices

If you have several devices that you want to remotely poll to collect event from, you can add `Device Address` lines to the previous script for each of the other devices you want to collect events from.

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
    <AddTo objPath="RemoteSources(Name=Domain Workstations)" >
        <Device Address="ws03.yourdomain.lab" Credentials="wcservice" />
        <Device Address="ws04.yourdomain.lab" Credentials="wcservice" />
        <Device Address="ws05.yourdomain.lab" Credentials="wcservice" />
        <Device Address="ws06.yourdomain.lab" Credentials="wcservice" />
```

```
    </AddTo>
</WinCollectScript>
```

This script adds four new remote devices to collect events from.

Add Active Directory lookup update script

Use this sample update script to add the Active Directory lookup values for your log source.

The Active Directory lookup converts GUIDs to real object names. This script updates the `Domain Controllers` remote source group and changes the `Security` source to include the `AD Domain Controller` and the `AD DNS` name to use to make the lookup.

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
    <Update objPath="RemoteSources(Name=Domain Controllers)/Source(Name=Security)" >
        <Parameter name="AD_Controller" value="yourdc.yourdomain.lab" />
        <Parameter name="AD_DNSDomain" value="youdomain.lab" />
        <Parameter name="AD_Lookup" value="true" />
    </Update>
</WinCollectScript>
```

Update script to add a secondary destination

Use this sample script to add a failover destination in case your primary destination fails.

For more information about secondary destinations, see "Adding a secondary destination" on page 37.

```
<?xml version="1.0" encoding="UTF-8"?>
<WinCollectScript version="10.0.1" >
    <Update objPath="Destinations/Destination(Name=QRadar)" setParam="Secondary"
value="qradarappliance.yourdomain.lab" />
</WinCollectScript>
```

Update script file warn and error messages

WinCollect accepts only files that begin with **update_*.xml**. Adding an `.xml` file that does not start with **update_** results in errors like this one in the **WinCollect.log** file:

```
01-31 13:58:04.350 WARN  Code.ChangeCtrl : Don't know what to do with this file: fakefile.xml --
only
```

```
update_*.xml files are considered update scripts
01-31 13:58:04.356 DEBUG Code.ChangeCtrl : Found 1 invalid files in the patch folder,
archiving...
01-31 13:58:04.357 WARN  Code.ChangeCtrl : There were no files in this patch affecting the
configuration.
01-31 13:58:04.359 ERROR Code.ChangeCtrl : Unable to move C:\Program Files\IBM\WinCollect\patch
\fakefile.xml to
C:\Program Files\IBM\WinCollect\patch_20200131_135804\fakefile.xml - error:2:The system cannot
find the file specified.
```

6
CHAPTER

WinCollect Sources

WinCollect Sources

IN THIS SECTION

A source is any Windows-based host that you configure WinCollect 10 to poll a log file or an event channel from.

WinCollect 10 can collect events from the following sources:

Microsoft Windows Event source

IN THIS SECTION

You can use the Microsoft Windows Event source to collect events from standard Event logs (Application, System, and Security), as well as application and services logs (XPath).

Table 14: Microsoft Windows Event source parameters

Parameter	Description
Type	Microsoft Windows Events
Channel	Select the channel that you would like to collect events from. Each channel that you want to collect from can be a unique source, or you can create an XPath query to collect from multiple channels. • Application • Security • System • "Forwarded Events (WEF)" on page 67 - When event subscriptions are configured, numerous Windows hosts can forward their events to this channel. • Directory Service • DNS Server • "XPath" on page 70 - XPath queries are structured XML expressions that you can use to retrieve events from standard logs or applications and service logs. XPath queries can also be used to filter out specific Event IDs.
Filter Enabled	You can use Pre-defined filters (such as NSA Filter) or other customer inclusion or exclusion filters.

Supported versions of Microsoft Windows Event

The WinCollect Microsoft Windows Event plug-in is not supported on versions of Microsoft Windows Event that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

MSEVEN6 protocol

The WinCollect 10 Microsoft Windows Event source uses the MSEVEN6 protocol by default. Use MSEVEN6 Protocol for all Windows Event collection unless directed otherwise by Juniper Customer Support. If you have a specific use case that requires MSEVEN, contact Juniper Customer Support for instructions on how to switch your source and to provide a description of the value of your MSEVEN use case.

Event filtering

IN THIS SECTION

You can configure the WinCollect 10 agent to include or exclude specific events that are collected from the Windows event log. Using event filtering, you can gather events that are of value to you while limiting the total events per second (EPS) that are sent to JSA.

You can configure WinCollect agents to ignore events globally by ID code or 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. You can separate multiple sources by using a semicolon. Event filters such as exclusion, inclusion, and NSA are available for the following source types:

- Security
- System
- Application
- DNS Server
- Directory Service
- Forwarded Events

The WinCollect agent requests all available events from the Event Collection API each time the value that is specified in the **Polling Interval** field expires. The agent then examines all of the events that are retrieved from the Event Collection API and ignores or includes events that match the filter. The agent then assembles the name=value pairs of the remaining events and forwards the events to either the JSA Console or the Event Collector appliance.

Event filtering configuration

WinCollect 10 no longer uses a separate field for inclusion or exclusion filters. The syntax that you use in the filter specifies whether you want to include or exclude events.

Exclusion filter The following example excludes event IDs 7000, any in the 7022-7026 range, 7031-7034 range, and 7045:

```
-(7000,7022-7026,7031-7034,7045)
```

Inclusion filter The following example includes event IDs 7000, any in the 7022-7026 range, 7031-7034 range, and 7045:

```
7000,7022-7026,7031-7034,7045
```

NSA filtering The NSA filter is available as a predefined filter. You can select NSA Filtering in the predefined filters menu only if you selected Security, System, Application, or DNS Server as the channel.

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.

In this example, event IDs 200 - 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:

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

Forwarded events

IN THIS SECTION

To better understand what forwarded events are, it is helpful to understand how to configure and set up Windows Event Forwarding (WEF).

Windows Event Forwarding basics

Windows Event Forwarding (WEF) is a powerful log forwarding solution that is integrated within modern versions of Microsoft Windows. Detailed documentation of WEF is available on the Microsoft Documentation page. The following list is a summary of WEF:

- Windows Event Forwarding provides the ability to send event logs, either via a push or pull mechanism, to one or more centralized Windows Event Collector (WEC) servers.
- WEF is agent-free and relies on native components that are integrated into the operating system. WEF is supported for both workstation and server builds of Windows.
- WEF supports mutual authentication and encryption through Kerberos (in a domain) or can be extended through the usage of TLS (additional authentication or for non-domain-joined machines).
- WEF has a rich XML-based language to control which event IDs are submitted, suppress noisy events, batch events together, and configure submission frequency. Subscription XML supports a subset of XPath, which simplifies the process of writing expressions to select the events you're interested in.

What are the WEC server's limitations?

Three factors limit the scalability of WEC servers. The general rule for a stable WEC server on commodity hardware is "10k x 10k," meaning no more than 10,000 concurrently active WEF Clients per WEC server and no more than 10,000 events per second average event volume.

Disk I/O	The WEC server does not process or validate the received event, but rather buffers the received event and then logs it to a local event log file (EVTX file). The speed of logging to the EVTX file is limited by the disk write speed. Isolating the EVTX file to its own array or using high-speed disks can increase the number of events per second that a single WEC server can receive.
Network Connections	While a WEF source does not maintain a permanent, persistent connection to the WEC server, it does not immediately disconnect after it sends events. This means that the number of WEF sources that can simultaneously connect to the WEC server is limited to the open TCP ports available on the WEC server.
Registry size	For each unique device that connects to a WEF subscription, a registry key (corresponding to the FQDN of the WEF Client) is created to store bookmark and source heartbeat information. If this information is not pruned to remove inactive clients, this set of registry keys can grow to an unmanageable size over time.

- When a subscription has more than 1000 WEF sources connect to it over its operational lifetime (lifetime WEF sources), the Subscriptions node on the Event Viewer can become unresponsive for a few minutes, but will function normally afterward.
- At more than 50,000 lifetime WEF sources, Event Viewer is no longer an option and you must use `wecutil.exe` (included with Windows) to configure and manage subscriptions.
- At more than 100,000 lifetime WEF sources, the registry is no longer readable, and the WEC server might need to be rebuilt.

WinCollect configuration

After you configure Windows Event Forwarding, you can configure the WinCollect agent to collect WEF events:

- Install the WinCollect 10 agent on your Windows Event Collector (WEC) servers.
- Configure a **Windows Events (default)** source and select **Forwarded Events (WEF)** as the channel.
- Deploy your changes.

NOTE:

- The maximum EPS supported by the Agent in a WEF environment is 10,000 EPS.
- Although the WinCollect agent displays only a single source in the user interface, the source listens and processes events for potentially hundreds of event subscriptions. One 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 JSA.
- Forwarded events are displayed as **Windows Auth @ *<hostname>*** in the **Log Activity** tab.

Additional information

For more information about managing large Windows Event Collection implementations, see https://www.ultimatewindowssecurity.com/webinars/watch_get.aspx?Attach=1&Type=SlidesPDF&ID=1426.

For more information about using Windows Event Forwarding to help with intrusion detection, see https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection.

XPath

IN THIS SECTION

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.

When you add an Xpath source, you can select from the list of predefined queries or enter your own custom query. If you select a predefined query, the UI adds that query to the **Xpath** field.

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 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.

NOTE: You must log in as an administrator to use Microsoft Event Viewer.

1. Click **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**.

   TIP: 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 severity of events that you want to include in your custom view.
7. Select an event source from the Event sources menu, or browse to a source from the **Event logs** menu.

8. Type the event IDs to filter from the event or log source.

   TIP: 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 username 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. Create a Windows Event Source with an XPath Channel and paste the XPath into the UI.
    - Using more than 10 XPath queries can affect WinCollect performance, depending on the XPath and the number of events that are coming into each channel.
    - Filtering events by a time range can lead to errors in collecting events.

XPath Examples

IN THIS SECTION

The following examples describe XPath queries you can use in WinCollect 10 to retrieve customized events from the Windows event logs.

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>
```

Retrieving Sysinternals Sysmon events

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>
```

Monitoring events for a specific user

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

```
<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>
```

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 >= 4776 and EventID <=
4777) )]]</Select>
  </Query>
</QueryList>
```

Table 15: Examples of event IDs used in credential logon

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

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 >= 4722
and EventID <= 4726) or (EventID >= 4741 and EventID
<= 4743) )]]</Select>
</Query>
</QueryList>
```

Table 16: Examples of event IDs used in credential logon

Event 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.

Microsoft IIS Source

IN THIS SECTION

The Microsoft Internet Information Services (IIS) source monitors IIS logs, which contain information about demographics and usage of the IIS web server.

Table 17: Microsoft IIS source parameters

Parameter	Description
Type	Microsoft IIS
Root directory	Default Value = **C:\inetpub\logs\LogFiles** **NOTE**: You no longer need to enter the UNC path for remote sources.
Logs	• Website (W3C) logs • File Transfer Protocol (FTP) logs • Simple Mail Transfer Protocol (SMTP) logs • Network News Transfer Protocol (NNTP) logs **TIP**: 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 WinCollect Microsoft IIS plug-in is not supported on versions of Microsoft IIS that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Microsoft Exchange Server source

IN THIS SECTION

You can configure the Microsoft Exchange Server source to monitor Message Tracking, SMTP, and Outlook Web Access (OWA) logs.

Message Tracking logs The message tracking log is a detailed record of all activity as mail flows through the pipeline on Mailbox servers and Edge Transport servers. You can use message tracking for message forensics, mail flow analysis, reporting, and troubleshooting.

SMTP logs Transport logs provide information about what's happening in the transport pipeline.

OWA logs Outlook Web Access logs are typical Internet Information Services (Exchange Server) logs.

Table 18: Microsoft Exchange Server source parameters

Parameter	Description
Type	Microsoft Exchange Server
Root directory	Default Value = **C:\Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs** **NOTE**: You no longer need to enter the UNC path for remote sources.
Message Tracking directory	Default Value = **MessageTracking** This is the name of the folder under the root directory that contains the Message Tracking log files.
SMTP directory	Default Value = **Hub\ProtocolLog** This is the name of the folder under the root directory that contains the SMTP log files.

Table 18: Microsoft Exchange Server source parameters *(Continued)*

Parameter	Description
OWA directory	Default Value = **C:\inetpub\logs\LogFiles** This is the name of the folder under the root directory that contains the OWA log files.

Supported versions of Microsoft Exchange Server

The WinCollect Microsoft Exchange Server plug-in is not supported on versions of Microsoft Exchange Server that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Microsoft DHCP Server source

IN THIS SECTION

The Microsoft DHCP Server source monitors DHCP logs, which contain information on successful or failed lease grants, depletion of the server's IP pool, or requests for messages and their corresponding acknowledgments.

Table 19: Microsoft DHCP Server source parameters

Parameter	Description
Type	Microsoft DHCP Server
Root directory	Default Value = **C:\Windows\System32\DHCP** **NOTE**: You no longer need to enter the UNC path for remote sources.

Table 19: Microsoft DHCP Server source parameters *(Continued)*

Parameter	Description
Log type	**IPV4** Include IPV4 files with file name pattern **DhcpSrvLog-*.log**. **IPV6** Include IPV6 files with file name pattern **DhcpV6SrvLog-*.log**. 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.

Supported versions of Microsoft DHCP Server

The WinCollect Microsoft DHCP Server plug-in is not supported on versions of Microsoft DHCP Server that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Microsoft SQL Server source

IN THIS SECTION

The SQL Server 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 events to JSA.

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.

Table 20: Microsoft SQL Server source parameters

Parameter	Description
Type	Microsoft SQL Server
Root directory	Default Value = **C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Log** **NOTE**: You no longer need to enter the UNC path for remote sources.

Supported versions of Microsoft SQL Server

The WinCollect Microsoft SQL Server plug-in is not supported on versions of Microsoft SQL Server that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Microsoft NPS source

IN THIS SECTION

The Microsoft NPS source monitors NPS logs, which contain user authentication and accounting requests.

Table 21: Microsoft NPS source parameters

Parameter	Description
Type	Microsoft NPS

Table 21: Microsoft NPS source parameters *(Continued)*

Parameter	Description
Root directory	Default Value = **C:\Windows\System32\LogFiles\NPS** **NOTE**: You no longer need to enter the UNC path for remote sources.

Supported versions of Microsoft NPS

The WinCollect Microsoft NPS plug-in is not supported on versions of Microsoft NPS that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Microsoft Forefront TMG source

IN THIS SECTION

Microsoft Forefront Threat Management Gateway installations create individual firewall and web proxy event logs in a common log directory. To collect these events with WinCollect 10, you must configure your Microsoft Threat Management Gateway to write event logs to a directory.

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

WinCollect 10 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:

Required field	Description
Client IP (c-ip)	The source IP address.
Action	The 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 username (cs-username)	The user account that made the data request of the firewall service.
Client username (username)	The user account that made the data request of the web proxy service.

Table 22: Microsoft Forefront TMG source parameters

Parameter	Description
Type	Microsoft Forefront TMG
Root directory	Example: **<Program Files>\<Forefront Directory>\ISALogs** **NOTE**: You no longer need to enter the UNC path for remote sources.
Log types	• W3C web protocol logs • W3C Firewall protocol logs • IIS web protocol logs • IIS Firewall protocol logs

Supported versions of Microsoft Forefront TMG

- Microsoft Forefront Threat Management Gateway 2010

Microsoft DNS Debug source

IN THIS SECTION

The Microsoft DNS Debug source monitors DNS Debug logs, which provide detailed data about all DNS information that is sent and received by the DNS server. This data is similar to the data that can be gathered using packet capture tools such as network monitor.

NOTE: 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.

Table 23: Microsoft DNS Debug source parameters

Parameter	Description
Type	Microsoft DNS Debug
Root directory	Default Value = No default value, as the file location is configured when you set up DNS Server debugging. **NOTE**: You no longer need to enter the UNC path for remote sources.
Filename pattern	The regular expression (regex) required to match the DNS debug log file set in the DNS manager. For Example, **dnslog.log**

Table 23: Microsoft DNS Debug source parameters *(Continued)*

Parameter	Description
File reader type	Reads the contents of the file. Both options have basic Unicode encoding support for byte-order marks. **Text (file held open)** — WinCollect maintains a shared read and write lock on the monitored log file. **Text (file open when reading)** — WinCollect maintains a shared read and write lock on the log file only when it reads the file.
Include details	Select this option if you have configured **Details** in the **Other options** section when configuring the logging.

Supported versions of Microsoft DNS Debug

The WinCollect Microsoft DNS Debug plug-in is not supported on versions of Microsoft DNS Debug that are designated end-of-life by Microsoft. After the software is beyond the Extended Support End Date, the product might still function as expected. However, Juniper does not make code or vulnerability fixes to resolve WinCollect issues for older software versions.

Enabling DNS debugging on Windows Server

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

The DNS role must be installed on the Windows Server.

1. Open the **DNS Manager** by typing 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. Enter the File path and name, and Maximum size.

NOTE: The File path and name, need to align with the Root Directory and File Pattern you provided when you configured the Microsoft DNS source.

6. Click **Apply** and **OK**.

Netapp Data ONTAP source

IN THIS SECTION

The NetApp Data ONTAP source monitors audit event logs (.evtx) of file and folder access attempts.

Table 24: Netapp Data ONTAP source parameters

Parameter	Description
Type	Netapp Data ONTAP
Channel	Security is the default.
Root directory	Directory where NetApp .evtx files are located.

Supported versions of Netapp Data ONTAP

- NetApp Data ONTAP 9.x

File Forwarder source

The File Forwarder source monitors many types of logs that are not covered as part of the standard WinCollect plug-ins. You can monitor logs continuously (Continuous Monitoring), or you can scan a folder for new files, process the contents, and wait for the next file (File Drop).

TIP: Because these logs fall outside of the standard plug-ins, there is no DSM to parse the events in JSA. You must either create a custom DSM or use the Universal DSM.

Table 25: File Forwarder source parameters

Parameter	Description
Type	File Forwarder
Root directory	Directory where the log files that you want to pull data from are stored. **NOTE**: You no longer need to enter the UNC path for remote sources.

Filename pattern	Only files that match this pattern are considered. This is an OS file filter.
	`*.* Will match all files`
	`*.log will match all files with a .log extenstion`
	`Server*.log will match all files with Server to start with and have.log extenstion`

Monitor subdirectories	Select if you would like the agent to monitor subdirectories of the root directory.
Monitoring algorithm	• **Continuous Monitoring** is intended for log files where data is continuously appended to the end of the log file. • **File Drop** is intended for log files that are "dropped" into the root log directory, read one time, and then ignored in the future.

CHAPTER 7

Advanced Settings

Advanced settings

IN THIS SECTION

To simplify the WinCollect 10 user interface, many finer control settings are included in the advanced UI.

Advanced users who want finer control of their WinCollect 10 agents can turn on the advanced UI by clicking the **Settings** icon, then moving the **Advanced UI** switch to the **On** position.

NOTE: Use the **Advanced UI** options only under direction from Juniper Customer Support.

Agent advanced settings

Advanced users who want finer control of their WinCollect 10 agents can access the following settings by turning on the advanced UI.

Parameter	Default value	Description
Notes		A free-form text field where you can add notes.

(Continued)

Parameter	Default value	Description
Identifier	The default value is the Agent Name.	The unique ID of this agent overrides the agent name. Changing this setting updates the payload header for the Agent messages. For example, if you add an identifier of `PERF-1-2019_IIS_Server`, the payload would change to `<13>Aug 04 15:32:57 PERF-1-2019_IIS_Server` JSA eventually adds this agent as a new source.
Status-Server		
Heartbeat custom properties		A comma-separated list of `keyword=value` to add to the heartbeat message. For example, if you add `department=Accounting, group=AC105` to this list, these values are added to the end of the Agent's Status heartbeat message. **Example:** `msg=ApplicationHeartbeat department=Accounting group=AC105`

Source advanced settings

IN THIS SECTION

You can use the following advanced settings to fine tune your WinCollect 10 sources.

Microsoft Windows events sources

"Microsoft Windows events advanced settings" on page 90

"EVTX Forwarder advanced settings" on page 91

File-based sources

"Common file-based plugin advanced settings" on page 94

"File Forwarder advanced settings" on page 96

"Microsoft DHCP Server advanced settings" on page 98

"Microsoft DNS Debug advanced settings" on page 99

"Microsoft Exchange Server advanced settings" on page 100

"Microsoft Forefront TMG advanced settings" on page 103

"Microsoft IIS advanced settings" on page 107

"Microsoft NPS advanced settings" on page 109

"Microsoft SQL Server advanced settings" on page 112

Microsoft Windows events advanced settings

You can use the following advanced settings to fine tune Microsoft Windows events sources.

Microsoft Windows events advanced settings

Parameter	Default value	Description
Identifier Override	hostname/IP	You can override the device identifier for this source
Tuning Profile	• Automatic Tuning • Low Event Rate • Medium Event Rate • High Event Rate • Max Event Rate • Manual Tuning	**Automatic tuning** Determines how to poll for events automatically and adjusts itself over time **Low event rate** Less than 1 event per minute, poll every 10 minutes, 100 events at a time. **Medium event rate** Less than 10 events per second, poll every 30 seconds, 200 events at a time. **High event rate** Less than 500 events per second, poll every 3 seconds, 2000 events at a time. **Max event rate** More than 500 events per second, poll continuously, 5000 events at a time. **Manual Tuning** Manually set the polling interval, events per pass, and batch size.
Manual Tuning		
• Polling Interval		The length of time (milliseconds) between polls.
• Events per pass		Maximum events to collect at each polling interval.
• Events per batch		Number of events to fetch per call to the source.

(Continued)

Parameter	Default value	Description
Event Levels	• Critical • Error • Warning • Information • Verbose • Always	• Include Critical events (level 1) • Include Warning events (level 3) • Include Verbose events (level 5) • Include Error events (level 2) • Include Information events (level 4) • Include Always logged events (level 0)
Keywords	• Audit Failure • Audit Success • Response Time • Classic	• Include keyword `0x10 0000 0000 0000` only for security events • Include keyword `0x20 0000 0000 0000` only for security events • Include keyword `0x01 0000 0000 0000` • Include keyword `0x80 0000 0000 0000` for events raised by using the **RaiseEvent**
SID Translation	Enabled	
Active Directory (AD) lookup	Not enabled	Turn the conversion of GUIDs into text on or off.
AD DNS domain name		
AD domain controller name		

EVTX Forwarder advanced settings

You can use the following advanced settings to fine tune EVTX Forwarder sources.

EVTX Forwarder advanced settings

Parameter	Default value	Description
Identifier Override	hostname/IP	You can override the device identifier for this source.
Filename pattern	***.evtx**	Only files that match this pattern are considered; this is an OS file filter.
Agent Device Type	WindowsLog	The **AgentDevice** field in the payload header.
Tuning Profile	• Automatic Tuning • Low Event Rate • Medium Event Rate • High Event Rate • Max Event Rate • Manual Tuning	**Automatic tuning** — Determines how to poll for events automatically and adjusts itself over time **Low event rate** — Less than 1 event per minute, poll every 10 minutes, 100 events at a time. **Medium event rate** — Less than 10 events per second, poll every 30 seconds, 200 events at a time. **High event rate** — Less than 500 events per second, poll every 3 seconds, 2000 events at a time. **Max event rate** — More than 500 events per second, poll continuously, 5000 events at a time. **Manual Tuning** — Manually set the polling interval, events per pass, and batch size.

Manual Tuning

• Polling Interval	The length of time (milliseconds) between polls.
• Events per pass	Maximum events to collect at each polling interval.
• Events per batch	Number of events to fetch per call to the source.

(Continued)

Parameter	Default value	Description
Event Levels	• Critical • Error • Warning • Information • Verbose • Always	• Include Critical events (level 1) • Include Warning events (level 3) • Include Verbose events (level 5) • Include Error events (level 2) • Include Information events (level 4) • Include Always logged events (level 0)
Keywords	• Audit Failure • Audit Success • Response Time • Classic	• Include keyword `0x10 0000 0000 0000` only for security events • Include keyword `0x20 0000 0000 0000` only for security events • Include keyword `0x01 0000 0000 0000` • Include keyword `0x80 0000 0000 0000` for events raised by using the **RaiseEvent**
Filter enabled	Checkbox	Turn the filter on or off.
• Predefined Filters		No Description
• Filter		An Event filter
SID Translation	Enabled	
Active Directory (AD) lookup	Not enabled	Turn the conversion of GUIDs into text on or off.
AD DNS domain name		
AD domain controller name		

(Continued)

Parameter	Default value	Description
Use Event Channel	Not enabled	Use the event's channel when available, and use **Channel** as the default.

Common file-based plugin advanced settings

You can use the following advanced settings to fine tune many file-based sources.

Common file-based plugin advanced settings

Parameter	Default value	Description
Identifier Override	hostname/IP	You can override the device identifier for this source.
Filename pattern	*.*	Only files that match this pattern are considered; this is an OS file filter.
Tuning Profile	• Automatic Tuning • Low Event Rate • Medium Event Rate • High Event Rate • Max Event Rate • Manual Tuning	**Automatic tuning** Determines how to poll for events automatically and adjusts itself over time **Low event rate** Less than 1 event per minute, poll every 10 minutes, 100 events at a time. **Medium event rate** Less than 10 events per second, poll every 30 seconds, 200 events at a time. **High event rate** Less than 500 events per second, poll every 3 seconds, 2000 events at a time. **Max event rate** More than 500 events per second, poll continuously, 5000 events at a time. **Manual Tuning** Manually set the polling interval, events per pass, and batch size.
Manual Tuning		
• Polling Interval		The length of time (milliseconds) between polls.
• Events per pass		Maximum events to collect at each polling interval.

(Continued)

Parameter	Default value	Description
• Events per batch		Number of events to fetch per call to the source.
Filter enabled	Checkbox	Turn the filter on or off.
• Predefined Filters		No Description
• Filter		An Event filter
File monitor type	• Auto • Notification based • Folder scanning	**Auto** Uses Notification based on the local system and Folder scanning (new files) on remote devices. **Notification based** Uses the Windows folder monitoring system. Not applicable on remote devices. **Folder Scanning** Scans a folder for new files created after the bookmark timestamp.
Files to Process	• New Files • All Files	Which files to accept when the system starts and there is no bookmark. **All Files** Scans a folder for new files created after the bookmark timestamp. If no bookmark exists, all files are processed. **New Files** Scans a folder for new files created after the bookmark timestamp. If no bookmark exists, all files are processed.
File reader type	• Text (file opened when reading) • Text (file held open)	The type of file reader that should be used to read the contents of the file.

(Continued)

Parameter	Default value	Description
File reader encoding	• UTF8 (no conversion) • ANSI (convert to UTF8) • UTF16 (convert to UTF8	How to deal with files without a BOM.

File Forwarder advanced settings

You can use the following advanced settings to fine tune File Forwarder sources.

File Forwarder advanced settings

Parameter	Default value	Description
File stale duration (minutes)	1440	In Continuous mode, how many minutes to keep monitoring a file that has not changed, before dropping it. The default value is one day. Select zero to keep monitoring files indefinitely.
Scan folder interval (seconds)	300	How often to scan the folder when **File Monitor Type** is Folder Scanning.
File parser	• One line per event • Multiple lines per event	How to read lines in the files.
Filter to accept lines		How many lines a mask or filter accepts lines to process. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
Multiple lines per event		

(Continued)

Parameter	Default value	Description
• Filter to start a multiline block		A mask or filter finds the first line to process in multilink. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• Filter to end a multiline block		A mask or filter finds the last line to process in multiline. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• Include the start and end lines of a multiline block		You can include or exclude the first (header) and last (footer) lines of a block.
• BlockTrimLines		Trims the lines before adding to the block separated by a space.
File		
• Filename pattern		A mask or filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• File_format		The **AgentLogFormat** payload header field. If you don't want to include this field, leave the value empty.
• File_fldList		The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab the list of fields from file header.
• File_fldListPrefix		The field list line prefix.

(Continued)

Parameter	Default value	Description
• File_fldListSep		The field list separator handles \t, XML, KEYED (CSV), or NCSA.

Microsoft DHCP Server advanced settings

You can use the following advanced settings to fine tune Microsoft DHCP Server sources.

Parameter	Default value	Description	Hidden
IPV4	checkbox		Yes
• Filename pattern	**DhcpSrvLog-*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.	
• IPV4_fldListPrefix	ID,|Identyfikator,	The field list line prefix.	
• IPV4_fldListSep	,	The field list separator handles \t, XML, KEYED (CSV), or NCSA.	
• IPV4_maxHdrLines	40	The maximum number of lines in the file header.	
IPV6			Yes
• Filename pattern	**DhcpV6SrvLog-*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.	
• IPV4_fldListPrefix	ID,|Identyfikator,	The field list line prefix.	

(Continued)

Parameter	Default value	Description	Hidden
• IPV4_fldListSep	,	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.	
• IPV4_maxHdrLines	40	The maximum number of lines in the file header.	

Microsoft DNS Debug advanced settings

You can use the following advanced settings to fine tune Microsoft DNS Debug sources.

Parameter	Value	Description
Include details	Checkbox	Turn on to include the details for each record. Uses the Multi Line File Parser.
OLD		
• Filename pattern		A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• OLD_fldList		The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• OLD_fieldCnt		Minimum-maximum. Auto classify file class using number of fields found when using **#FIELDS** or **#HRD_LINES**.
• OLD_maxHdrLines		The maximum number of lines in the file header.

(Continued)

Parameter	Value	Description
NEW		
• Filename pattern		A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• NEW_fldList		The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• NEW_fieldCnt		Minimum-maximum. Auto classify file class using number of fields found when using **#FIELDS** or **#HRD_LINES**.
• NEW_maxHdrLines		The maximum number of lines in the file header.

Microsoft Exchange Server advanced settings

You can use the following advanced settings to fine tune Microsoft Exchange Server sources.

Microsoft Exchange Server advanced settings

Parameter	Default value	Description
W3C		
• Filename pattern	**ex*.log|u_ex*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• W3C_format	**W3C**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• W3c_fldList	**#FIELDS**	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
IIS		
• Filename pattern	**in*.log|u_in*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• IIS_format	**IIS**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• IIS_fldList	`ClientIP,Username,Date,Time,Service,ServerName,ServerIP,TimeTaken,ClientBytesSent,ServerBytesSent,ServiceStatus,WindowsStatus,RequestType,Target,Parameters`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
NCSA		
• Filename pattern	**nc*.log|u_nc*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• NCSA_format	**NCSA**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• NCSA_fldList	`RemoteHost,RemoteLog,Username,Timestamp,GMTOffset,` `Method,Resource,Protocol,ServiceStatus,BytesSent`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
Message Tracking		
• Filename pattern	**MSGTRK*.LOG|msgtrk*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• MSGTRK_format	**MSGTRK**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• MSGTRK_fldList	**#FIELDS**	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• MSGTRK_fldListSep	,	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.
SMTP Tracking		
• Filename pattern	**RECV*.LOG|SEND*.LOG**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• SMTP_format	**SMTP**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• SMTP_fldList	**#FIELDS**	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• SMTP_fldListSep	,	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.

Microsoft Forefront TMG advanced settings

You can use the following advanced settings to fine tune Microsoft Forefront TMG sources.

Microsoft Forefront TMG advanced settings

Parameter	Default value	Description
W3C Web protocol logs		
• Filename pattern	***WEB*.w3c**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• W3C_WEB_format	**W3C-WebProxy**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• W3C_WEB_fldListSep	`\t`	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.
W3C Firewall protocol logs		
• Filename pattern	***FWS*.w3c**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.

(Continued)

Parameter	Default value	Description
• W3C_FWS_format	**W3C-Firewall**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• W3C_FWS_fldListSep	`\t`	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.
IIS Web protocol logs		
• Filename pattern	***WEB*.iis**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• IIS_WEB_format	**IIS-WebProxy**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• IIS_WEB_fldList	`Client IP,Client Username,Client Agent,Authenticated Client,Log Date,Log Time,Service,Server Name,Referring Server,Destination Host Name,Destination IP,Destination Port,Processing Time,Bytes Received,Bytes Sent,Protocol,Transport,HTTP Method,URL,MIME Type,Object Source,Result Code,Cache Info,Rule,Filter Information,Source Network,Destination Network,Error info,Action,GMT Log Time,Authentication Server,NIS Scan Result,NIS Signature,Threat Name,Malware Inspection Action,Malware Inspection Result,URL Category,Content Delivery Method,UAG Array Id,UAG Version,UAG Module Id,UAG Id,UAG Severity,UAG Type,UAG Event Name,UAG Session Id,UAG Trunk Name,UAG Service Name,UAG Error Code,Malware Inspection Duration (msec),Threat Level,Internal Service Info Log Fields,NIS Application Protocol,NAT Address,URL Categorization Reason`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• IIS_WEB_fldListSep	,	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.
IIS Firewall protocol logs		
• Filename pattern	*FWS*.iis	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.

(Continued)

Parameter	Default value	Description
• IIS_FWS_format	`IIS-Firewall`	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• IIS_FWS_fldList	`Server Name,Log Date,Log Time,Transport,Client IP and Port,Destination IP and Port,Original Client IP,Source Network,Destination Network,Action,Result Code,Rule,Protocol,Bidirectional,Bytes Sent,Bytes Sent Delta,Bytes Received,Bytes Received Delta,Processing Time,Processing Time Delta,Source Proxy,Destination Proxy,Client Host Name,Destination Host Name,Client Username,Client Agent,Session ID,Connection ID,Network Interface,Raw IP Header,Raw Payload,GMT Log Time,NIS Scan Result,NIS Signature,NAT Address,Forefront TMG Client FDQN,Forefront TMG Client Application Path,Firewall Client Application SHA1 Hash,Forefront TMG Client Application trust state,Forefront TMG Client Application Internal Name,Forefront TMG Client Application Product Name,Forefront TMG Client Application Product Version,Forefront TMG Client Application File Version,Forefront TMG Client Application Original File Name,Internal Service Info Log Fields,NIS Application Protocol,Forefront TMG Client Version`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.

(Continued)

Parameter	Default value	Description
• IIS_FWS_fldListSep	,	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.

Microsoft IIS advanced settings

You can use the following advanced settings to fine tune Microsoft IIS sources.

Microsoft IIS advanced settings

Parameter	Default value	Description
Monitor Subdirectories	Enabled	Monitoring of subfolders on `ON` by default for IIS.
FTP protocol logs	Disabled	
NNTP/News protocol logs	Disabled	
SMTP/Mail protocol logs	Disabled	
W3C protocol logs	Enabled	
W3C		
• Filename pattern	**ex*.log|u_ex*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• W3C_format	**W3C**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• W3c_fldList	**#FIELDS**	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
IIS		
• Filename pattern	**in*.log|u_in*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• IIS_format	**IIS**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.
• IIS_fldList	`ClientIP,Username,Date,Time,Service,ServerName,ServerIP,TimeTaken,ClientBytesSent,ServerBytesSent,ServiceStatus,WindowsStatus,RequestType,Target,Parameters`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
NCSA		
• Filename pattern	**nc*.log|u_nc*.log**	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• NCSA_format	**NCSA**	The `AgentLogFormat` payload header field. If you don't want to include this field, leave the value empty.

(Continued)

Parameter	Default value	Description
• NCSA_fldList	`RemoteHost,RemoteLog,Username,Timestamp,GMTOffset,` `Method,Resource,Protocol,ServiceStatus,BytesSent`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.

Microsoft NPS advanced settings

You can use the following advanced settings to fine tune Microsoft NPS sources.

Microsoft NPS advanced settings

Parameter	Default value	Description
DTS		
• DTS_fldList	`IASFmt:Static=DTS,Timestamp,Computer-Name,Event-Source,Class,User-Name,Acct-Session-Id,NAS-IP-Address,NAS-Identifier,NAS-Port,Calling-Station-Id,Client-IP-Address,Client-Vendor,Client-Friendly-Name,Proxy-Policy-Name,Provider-Type,SAM-Account-Name,Authentication-Type,NP-Policy-Name,Fully-Qualifed-User-Name,Quarantine-Update-Non-Compliant,Packet-Type,Reason-Code,*`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• DTS_fldListSep	**XML**	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.

(Continued)

Parameter	Default value	Description
• DTS_fldInclude	**SET**	**ALL** All fields in the list are included. **SET** Only fields that are set are included. **VALUE** Only fields with an actual value are included. A value of `0` is empty.
IAS		
• IAS_fldList	`IASFmt:Static=IAS,NAS-IP-Address,User-Name,Record-Date,Record-Time,Service-Name,Computer-Name,Fully-Qualified-User-Name:Key=4130,Client-IP-Address:Key=4108,NAS-Manufacturer:Key=4116,Client-Friendly-Name:Key=4128,Proxy-Policy-Name:Key=4154,Provider-Type:Key=4155,SAM-Account-Name:Key=4129,Class:Key=25,Authentication-Type:Key=4127,NP-Policy-Name:Key=4149,Quarantine-Update-Non-Compliant:Key=8136,Packet-Type:Key=4136,Reason-Code:Key=4142`	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• IAS_fldListSep	**KEYED**	The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.
ODBC		

(Continued)

Parameter	Default value	Description
• ODBC_fldList	IASFmt:Static=ODBC,Computer-Name,Service-Name,Record-Date,Record-Time,Packet-Type,User-Name,Fully-Qualified-User-Name,Called-Station-ID,Calling-Station-ID,Callback-Number,Framed-IP-Address,NAS-Identifier,NAS-IP-Address,NAS-Port,Client-Vendor,Client-IP-Address,Client-Friendly-Name,Event-Timestamp,Port-Limit,NAS-Port-Type,Connect-Info,Framed-Protocol,Service-Type,Authentication-Type,NP-Policy-Name,Reason-Code,Class,Session-Timeout,Idle-Timeout,Termination-Action,EAP-Friendly-Name,Acct-Status-Type,Acct-Delay-Time,Acct-Input-Octets,Acct-Output-Octets,Acct-Session-Id,Acct-Authentic,Acct-Session-Time,Acct-Input-Packets,Acct-Output-Packets,Acct-Terminate-Cause,Acct-Multi-Ssn-ID,Acct-Link-Count,Acct-Interim-Interval,Tunnel-Type,Tunnel-Medium-Type,Tunnel-Client-Endpt,Tunnel-Server-Endpt,Acct-Tunnel-Connection,Tunnel-Pvt-Group-ID,Tunnel-Assignment-ID,Tunnel-Preference,MS-Acct-Auth-Type,MS-Acct-EAP-Type,MS-RAS-Version,MS-RAS-Vendor,MS-CHAP-Error,MS-CHAP-Domain,MS-MPPE-Encryption-Types,MS-MPPE-Encryption-Policy	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• ODBC_fldListSep	,	The field list separator handles \t, XML, KEYED (CSV), or NCSA.

Microsoft SQL Server advanced settings

You can use the following advanced settings to fine tune sources.

Microsoft SQL Server advanced settings

Parameter	Default value	Description
SQL		
• Filename pattern	*	A mask/filter to categorize files into this file class. You can select * (many chars) **?** (one char), **#** (a number), or a regex.
• SQL_fldList	**Date,Time,Source,Message:Last=TakeRest**	The CSV fields to include in the payload. Use **#FIELDS** to grab first line in the file that starts with **#Fields**. Use **#HDL_LINES** to grab list of fields from file header.
• SQL_fldListSep		The field list separator handles `\t`, `XML`, `KEYED (CSV)`, or `NCSA`.

System advanced settings

You can use the following advanced settings to more accurately tune the system that contains the WinCollect 10 agent.

Table 26: WinCollect 10 system advanced settings

Parameter	Default value	Description
Max Archived Patch Folders	20	The maximum number of patch archive folders (**patch_*yyyymmdd_hhmmss***) to keep.
Configuration Console's Port	3000	The port that the agent's webserver hosts the configuration console on.

Table 26: WinCollect 10 system advanced settings *(Continued)*

Parameter	Default value	Description
Maximum Entity Read Buffer	16384	The size of the buffer to read large entities from the console UI.
Check patch folder every (seconds)	10	How often to check whether files are in the patch folder. Select **0** to disable.
Check for expired cached IP every (seconds)	60	How often to check whether cached IPs are expired. Select **0** to disable.
Check the health of the system every (seconds)	30	How often to check sources, targets, threads, and resources.
Source device retry wait time	60000	The initial wait time (ms) to try to reconnect to a source device. The default value is 1 minute.
Source device maximum retry amount	0	The number of times to try to reconnect to a source device before it gives up. Select **0** to keep trying indefinitely.
Source device exponential wait factor	5	The exponential factor to increase wait time after each try. The default setting of **5** tries at 5 minutes, 25 minutes, 125 minutes... Select **1** to use linear wait factor.
Source device linear wait factor	0	The linear factor to increase wait time after each try. Example: If the retry wait time is 300000 (5 minutes), and the wait factor is 1, the source tries at 5 minutes, 10 minutes, 15 minutes...
Source device maximum retry wait time	360000	The max wait time to try to reconnect to a source device. The default value is 1 hour.

Table 26: WinCollect 10 system advanced settings *(Continued)*

Parameter	Default value	Description
Bad SID Translation text	N/A	The text to use when a SID cannot be translated, use **IGNORE** to leave the SID as is.
Source start query delay time	1000	Delay in milliseconds to start querying the sources.
Source start duration	Auto	How long to take to start querying all the sources, **AUTO** queries up to 25 per second and evenly spread the load over 5 seconds.
Source minimum thread count	0	0 means auto calc (DvcCnt/100)
Source maximum thread count	0	0 means auto calc (DvcCnt/20)
Source batch size	100	
Source cached IP expiration time (seconds)	900	Seconds to keep the computed IP cached for each remote device. The default value is 15 minutes. Select **0** to disable refresh.
Destination maximum number of events in memory	10000	The maximum number of events in memory that are waiting to be sent before the agent starts dumping to disk. **NOTE**: This setting is per destination.
Destination device retry wait time	10000	The initial wait time (ms) to try to reconnect to a destination. The default value is 10 seconds.
Destination device maximum retry amount	0	The number of times to try to reconnect to a destination before it gives up. Select **0** to keep trying indefinitely.

Table 26: WinCollect 10 system advanced settings *(Continued)*

Parameter	Default value	Description
Destination device exponential wait factor	1	The exponential factor to increase wait time after each try. Select **1** to use linear wait factor.
Destination device linear wait factor	10000	The linear factor (ms) to increase wait time after each try. Example: If the retry wait time is 10000 (10 seconds), and the wait factor is 1, the source tries at 10 seconds, 20 seconds, 30 seconds...
Destination device maximum retry wait time	300000	The maximum wait time (ms) to try to reconnect to a destination. The default value is 5 minutes.
Destination maximum payload size for UDP	1020	The maximum payload size for UDP.
Destination maximum payload size for TCP	32000	The maximum payload size for TCP/TLS.
Check Primary every (seconds)	60	How often to check whether the primary destination is back online when the agent is using the secondary destination.
Use non blocking socket connection	Enabled	Use non blocking socket connection works in combination with Socket connection timeout setting.
Socket connection timeout	10000	Milliseconds allowed to connect.
Socket connection established wait time	50	Milliseconds to wait for a connection to be established between checks.

Table 26: WinCollect 10 system advanced settings *(Continued)*

Parameter	Default value	Description
Allow live statistics collection	Enabled	While the console is running, the agent collects live stats for the Dashboard once per second.
Statistics heartbeat timeout	120	If the agent doesn't receive a request from the console after 120 seconds, it stops collecting live stats.
Minimum task count in queue for using index	24	The minimum number of tasks in the queue before the agent starts using an index.
Number of tasks per node in index	8	The number of tasks per node in the index.
Number of task requeue in index	20	The number of times a task is put back in the queue before the agent checks the average sequential search.
Maximum average number of sequential tasks	16	The maximum average number of sequential tasks to search per node before reindexing.
User data directory	%ALLUSERSPROFILE%/ WinCollect10Beta/Data/	The directory where WinCollect stores events, raw messages, and bookmarks.
Events on disk max disk space	6144	The maximum disk space to use (MB), when pushing events on disk when unable to send.
Max worker pool kill timeout	10000	The total time allocated to peaceful termination of the whole worker pool split equally between the number of workers in the pool.

Table 26: WinCollect 10 system advanced settings *(Continued)*

Parameter	Default value	Description
Max worker kill timeout	2000	The maximum timeout allowed per worker regardless of the pool maximum.

8
CHAPTER

The WinCollect 10 Statistics File

The WinCollect 10 statistics file

Every agent has a statistics file that you can use to see the events that the agent processed over an amount of time. The statistics file is stored in the `/logs` directory where the agent is installed.

You can open the statistics file in any text editor. The file looks similar to the following example:

```
From 20210915.130000 to 20210915.140000
Destination//QRadar:
3.5/162,6,4,3,6,31,4,3,3,3,26,3,3,3,6,30,3,4,3,3,32,3,4,4,7,2.4/116,3,3,3,3,25,4,3,3,6,31,4,3,3,3
,23,3,3,3,10,48,9,10,4,12,31,12,3,3,6,19,12,3,3,4
Source//Local//Application: 0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Source//Local//DNS Debug:
162,,,,,6,,,,,2,,,,,4,,,,,6,,,,,116,,,,,2,,,,,8,,,,,2,,,,,28,,,,,12,,,,,4,,,,
Source//Local//Security:
27,5,4,3,6,24,3,3,3,3,24,3,3,3,6,26,3,4,3,3,26,3,4,3,6,21,3,3,3,3,23,4,3,3,6,23,4,3,3,3,21,3,3,3,
10,20,9,6,4,12,15,12,3,3,6,15,12,3,3,4
Source//Local//XPath Sysmon Powershell:
3,1,,,,1,,,,,,,,,,,,,,,,,,,,1,2,,,,,,,,,,,,,,,,,,,,,,,2,,,4,,,,,,,,,,
StatusServer//10.10.218.221:
0,,,,1,,,,,1,,,,,1,,,,,1,,,1,,1,,,1,,1,,,,1,,1,,,,,1,,,,1,,1,,1,,,,1,,,,,1,,,,,1
UserData//EvtsOnDisk: 0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Table 27: How to read the statistics file

Value	Description
`From 20210915.130000 to 20210915.140000`	The statistics file is updated every minute and creates a new section every hour. In this example, the data is from Sept 15, 2021, from 1 PM to 2 PM.

Table 27: How to read the statistics file *(Continued)*

Value	Description
`Destination//QRadar: 3.5/162,6,4,3,6,31,4,3,3, 3,26,3,3,3,6,30,3,4,3,3,32,3,4,4,7, 2.4/116,3, 3,3,3,25,4,3,3,6,31,4,3,3,3,23,3,3, 3,10,48,9, 10,4,12,31,12,3,3,6,19,12,3,3,4`	This line contains an entry for each destination you are sending logs to. In this example, you have one destination that is named JSA. Events per Minute (EPM) are logged each minute. Therefore, this comma-separated line contains 60 entries. The most current entries are the values on the far right. Numbers in the X/Y format represent the average and highest EPS seen for that minute. • For example, `3.5/162` means that the average EPS was 3.5 and the most events that are processed during 1 second in that minute was 162.
`Source//Local//Application: 0,,,,,,,,,,,,,,, ,,,,,,,,,,,,,,, ,,,,,,,,,,,,,,,,,,,,,,,`	This is a source that is named **Application**, in the **Local** group. This is the source that is collecting events from the local Application event channel.
`Source//Local//DNS Debug: 162,,,,,6,,,,,2,,,,, 4,,,,,6,,,,,116,,,,,2,,,,,8,,,,,2,, ,,,28,,,,, 12,,,,,4,,,,`	This is a source that is named **DNS Debug** in the **Local** group. This source is collecting DNS Debug logs on the local machine.
`Source//Local//Security: 27,5,4,3,6,24,3,3,3, 3,24,3,3,3,6,26,3,4,3,3,26,3,4,3,6, 21,3,3,3,3, 23,4,3,3,6,23,4,3,3,3,21,3,3,3,10,2 0,9,6,4,12, 15,12,3,3,6,15,12,3,3,4`	This is a source that is named **Security**, in the **Local** Group. This is the source that is collecting events from the local Security event channel. As expected, this is the busiest source. The security channel typically generates the most traffic in the standard event logs.
`Source//Local//XPath Sysmon Powershell: 3,1,,,,1,,,,,,,,,,,,,,,,,,1,2,,,,, ,,,,,,,,, ,,,,,,,2,,,4,,,,,,,,`	This is a source that is named **XPath Sysmon Powershell**, in the **Local** group. This is the source that is collecting events from the Sysmon and PowerShell applications and services event logs.

Table 27: How to read the statistics file *(Continued)*

Value	Description
`StatusServer//10.10.218.221: 0,,,,1,,,,,1,,,,, 1,,,,,1,,,1,,1,,,1,,1,,,,1,1,,,,,1, ,,,1,1,1,,, ,1,,,,,1,,,,,1`	This is where the status messages are sent, and includes heartbeat messages and any service stop or start and Agent error messages. Typically, these have a very low EPS count (one message every 5 minutes).
`UserData//EvtsOnDisk: 0,,,,,,,,,,,,,,,,,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,,,,,,,,,,,,,,,,,`	This shows whether events are being stored on the disk. For example, the agent can't communicate to JSA and thus stores the events to disk until it can open the communication.

CHAPTER 9

WinCollect Terminology

WinCollect terminology

Definitions of terms commonly used in WinCollect 10 documentation.

Advanced installer	Select the **Advanced** option on the WinCollect graphical installer to specify which WinCollect 10 components to install and where to install them.
Advanced UI	Turn on this feature to see advanced settings in the WinCollect Console.
Agent settings	Use **Agent settings** to configure specific agent settings, such as the Agent Identifier and where to send status messages.
Destination	Destinations are where you want to send your event data. You can send syslog event data using UDP, TCP, or TLS protocols. A destination can be any JSA appliance in your deployment.
Identifier	The agent identifier is usually the hostname value from the environment settings.
Local source	A source that is configured to collect data that is local to where the agent is installed.
Log configuration	Increase or decrease the level of logging. For example, set logging to DEBUG.
Remote source	A source that is configured to collect data that is not local to where the agent is installed. A remote source requires credentials and a device name to collect the remote events.
Secondary destination	You can add a secondary destination to receive events from your WinCollect agents if the primary destination fails.
Service status	In the Service Status section of the console, you can quickly see the status of the WinCollect agent service.
Source	A source is similar to a log source, except that unlike some prior WinCollect log sources (such as Microsoft Security Event Logs), which collected data from one to many sources (Security, System, or Application event channels), WinCollect 10 uses single sources. For example, if you want to collect Security events, that is one source. Using sources, you can apply configuration changes at a lower level. For example, instead of collecting events for all channels at 3 seconds, you can configure different polling intervals based on how busy each channel is.
Stand-alone	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, or remote Windows hosts.
Status messages	WinCollect 10 can send agent status information (e.g., Service is stopping or starting) to JSA in the form of status messages.

Support files	The **Collect Support Files** option collects and compresses the necessary WinCollect configuration and log files and saves them to the log folder. You can then upload this file to a support case with Juniper Customer Support.
Update script	You can use update scripts to change the agent configuration without making manual changes to the **AgentConfig.xml** file or using the Configuration console. When you copy an update script to the WinCollect patch directory, the agent completes the actions described in your script.