



TopView Failover / Redundancy Configuration

View the latest version of this document:

If this document is available through your TopView installation or was previously downloaded you can view the latest version [here](#).

Last update: June 5, 2023

Recent updates:

- June 5th, 2023
 - Added additional conditions for topview_engine_ok available in v7.1
- January 9th, 2023
 - DataPath syncing and file sharing for OPCUA
- December 15th, 2022
 - Repointing path to license.ini using utility in TopView v7.0 and later
 - Configuration of non-Engine TopView Services on the secondary machine (Web Configurator and Message Receiver)
- April 15th, 2022
 - Information on using Watchdog TCP Server and Watchdog tags for redundancy instead of Performance Counters or health tags. (TopView 6.37 and later)
- January 10th, 2022
 - Added warning for shared DataPath: do not use a mapped drive letter to access the share
- October 31, 2021
 - Sync DataPath folders: add link to Syncing support KB, remove Microsoft Sync Toy which is no longer available
- August 10th, 2021
 - Permission to read remote performance counters: user account should be a member of 'Performance Log Users' in addition to 'Administrators' and 'Performance Monitor Users'
- February 15, 2021
 - For the shared DataPath solution, added additional steps to rename the local datapath folders and to verify that both machines are correctly pointed to the share location
- December 14, 2020

- Updated logic function for 3-tag solution
- October 1, 2020
 - Added additional files/folders to sync when using a local DataPath
- September 16, 2020
 - Added 3-tag configuration for failover
 - Added Engine start/stop options available in TopView 6.35
- September 4, 2020
 - Synchting added as a recommended folder synchronization tool

©Copyright 2023 Exele Information Systems, Inc.

Exele Information Systems

+1 (585) 385-9740

<http://www.exele.com>

Support: support@exele.com

Sales: sales@exele.com

Contents

Overview of TopView Failure and Redundancy	5
What is a TopView Failure?.....	5
What is TopView Redundancy?	5
Licensing for TopView Redundancy	5
Terminology	6
Getting ready for TopView Redundancy.....	7
Licensing.....	7
TopView Version	7
DataPath and configuration sharing	7
Option 1: Synchronize the primary and secondary DataPath configuration information.....	8
Option 2: Share the DataPath	9
Non-Engine Services	14
Web Configurator	14
Messaging Receiver	15
WatchDog monitoring: Watchdog tags, Health tags, or Performance Counters?	16
Watchdog Monitoring: Watchdog tags	17
WatchDog Monitoring: Performance Counters.....	18
WatchDog Monitoring: Health tags	19
Configure TopView Redundancy using Watchdog Tags.....	20
Primary machine	20
Configure the Watchdog Server for each Primary Engine	20
Enable reading Watchdog tags from remote machines (firewall)	21
Secondary machine.....	22
TopView license for Watchdog Engine	22
Install TopView Engine Services.....	22
Create the WatchDog TopView Engine.....	23
Configure TopView Redundancy using Performance Counters.....	27
Primary machine	27
Configure Primary Engines to write Performance Counters.....	27
Enable reading Performance Counters from remote machines	28
Secondary machine.....	29
TopView PerfMon license	29

Install TopView Engine Services	29
Create the WatchDog TopView Engine.....	31
Configure TopView Redundancy using Health Output Tags	41
Primary machine	41
Configure Primary Engines to write to health output tags	41
Secondary machine.....	43
Additional Failover/Redundancy components	45
VOIP Callout Failover – Grandstream IP PBX	45
One shared Grandstream device	45
Two different Grandstream devices	45
OPCUA Configuration Files.....	46
Overview	46
Sharing OPCUA information.....	47

Overview of TopView Failure and Redundancy

What is a TopView Failure?

A “TopView failure” is defined as the condition where one or more TopView Engines have either stopped running or are unavailable. This failure condition may be caused by:

- Stopping a running TopView Engine instance/configuration
- Shutdown of the TopView machine
- Failure of the network between TopView and other systems

What is TopView Redundancy?

TopView Redundancy, also called “TopView Failover Configuration” or “TopView Backup Configuration”, allows a second machine to monitor the health of the TopView Engines on the primary machine. When one or more TopView Engines on the primary machine fail, the second machine runs a copy of the primary TopView Engines until the primary Engines are again available.

Health monitoring of the primary Engines is performed using a WatchDog Engine on the secondary machine. The WatchDog monitoring of the primary Engines can be accomplished through heartbeat tags written by the primary machine (if your data source supports outputs) or through Windows Performance Counters.

Licensing for TopView Redundancy

You must purchase a failover/backup TopView license before configuring TopView Redundancy.

Terminology

- **Machine:** a physical computer or virtual machine
- **Engine:** A TopView Engine typically running as a Windows Service. The behavior of each TopView Engine is controlled by a TopView configuration file.
- **Primary machine:** the main TopView machine running one or more TopView Engines. If you are running TopView without redundancy this is your current TopView machine.
- **Secondary machine:** the failover TopView machine running the WatchDog Engine and, during a time of failover, copies of the primary TopView Engines
- **WatchDog Engine:** a TopView Engine on the secondary machine that is monitoring the health of one or more TopView Engines on the primary machine. It can start and stop copies of the primary Engines based on the health of each primary Engine. The WatchDog can also be configured to send notifications regarding failures.
- **%windir%:** the location of the Windows root folder, usually C:\Windows
- **%ProgramData%:** the location of the machine's program data folder, usually C:\ProgramData
- **DataPath:** the TopView DataPath where all writable files are stored. This includes configuration files (DataPath\Config) and log files (DataPath\Log). The location of the TopView DataPath is defined in the file %windir%\TopView.ini and the DataPath folder can be opened from the TopView Configurator (top menu, Tools...Open DataPath folder). The default location is %ProgramData%\Exele\TopView (usually C:\ProgramData\Exele\TopView)

Getting ready for TopView Redundancy

Licensing

You must purchase a TopView Failover/Backup license in addition to the TopView license used on your primary machine. Please [contact us](#) if you are unsure of your current license or if you would like to purchase a failover/backup license.

TopView Version

The same version of TopView should be installed on the primary and secondary machines

DataPath and configuration sharing

Each TopView Engine is controlled by a TopView configuration file stored in DataPath\Config. This folder also contains global configuration settings such as Schedules and Contacts. The DataPath\TopView.ini file contains Global Options that may have been configured.

During failover the secondary machine will start copies of the Engines that are normally running on the primary machine. Therefore the secondary machine should have access to the same configuration information used by the Engines on the primary machine.

The default location for the DataPath is %ProgramData%\Exele\TopView (C:\ProgramData\Exele\TopView) which is on the local machine of the primary and secondary machine. Unless you have copied the files and folders from the primary to the secondary machine the configuration information from the primary machine is unavailable to the secondary machine during failover. ***Redundancy requires that both the primary and secondary machine have access to the same configuration information.***

Two options exist for DataPath and configuration sharing between the primary and secondary machines:

1. Leave the DataPath for the primary and secondary machines as their default location (local to each machine) and configure a synchronization process from the primary to the secondary machine.
2. Share the same DataPath for both machines. Most users will find that this is the better option. There are a few ways to do this:
 - a. Move the DataPath to a 3rd machine or device that has a share visible to the primary and secondary machine.
This is the recommended method.
 - b. Use the local DataPath on the secondary machine by creating a share that is visible from the primary machine. The secondary machine will see the datapath as c:\... and the primary machine will see the datapath as \\sharename\...
This method requires that the secondary machine is always available.
 - c. Use the local DataPath on the primary machine by creating a share that is visible from the secondary machine. The primary machine will see the datapath as c:\... and the secondary machine will see the datapath as \\sharename\...
This method requires that the primary machine is always available.

[Option 1: Synchronize the primary and secondary DataPath configuration information](#)

This option requires that you perform folder synchronization to ensure that the secondary machine's DataPath contains the same configuration folders and files as the primary machine's DataPath.

Which files/folders should you sync?

- **File: TopView.INI**
Global settings configured through Global Setting>Global Options in the TopView Configurator
- **Folder: Config**
Engine configuration files, Contacts, Notification Groups, Logic Functions, ...
OPCUA users: exclude theConfig\OpcUA folder. See "OPCUA Configuration Files" for more information.
- **Folder: Sounds**
Audible Alarm sound files
- **Folder: Voice**
Voice Notification sound files (greetings, acknowledge confirmation, ...)

How to sync

TopView does not provide any folder/file synchronization tools so you will need to consider a 3rd party tool to perform this function. Although we do not recommend or endorse synchronization tools you may want to research the following tools or search for similar tools:

- [Syncthing](#) (see our [support KB](#) on Syncthing recommendations)
- [SyncBack from 2BrightSparks](#) (non-server OS)
- [FreeFileSync](#) (server OS)

Option 2: Share the DataPath

This option allows the primary and secondary machine to share the same DataPath, removing the need to synchronize configuration settings.

The following details assume that the shared DataPath is placed on a 3rd machine that is different than the primary or secondary machine.

To share the DataPath you will need to perform the following steps:

1. Create the network share for the new DataPath.
Example: create a share named [\\NetWorkDevice\TopViewDataPath](#)
Warning: do not use a mapped drive letter to access the share from the primary or secondary machine. Use the UNC path \\machinename\sharename
2. Stop all TopView applications on the primary and secondary machines. This includes any TopView Engines Services that may be running.
3. Determine which machine contains the configuration files that will be moved to the new shared location - normally this will be the primary machine's DataPath located in C:\ProgramData\Exele\TopView of the primary machine.
4. Copy **all of the files and folders** from the DataPath determined in #3 to the shared location
Example: Copy all files and folders from
C:\ProgramData\Exele\TopView on the primary machine
to
[\\NetWorkDevice\TopViewDataPath](#)
5. Reprint the DataPath of the primary and secondary machines to the new shared location.
On the primary and secondary machines:
 - a. Edit the file %windir%\TopView.ini (usually C:\Windows\TopView.ini) using Notepad or another text file editor. Note: you may need to run the text editor "as Administrator" before you can save changes to this file.
 - b. Change the line "DataPath=" to point to the new shared location
Example:
Change
DataPath=C:\ProgramData\Exele\TopView
to
DataPath=\\NetWorkDevice\TopViewDataPath
 - c. **Warning:** do not use a mapped drive letter to access the share from the primary or secondary machine. Use the UNC path \\machinename\sharename
 - d. Save the %windir%\TopView.ini file

6. Repoint the location of the license.ini file for the primary and secondary machines

The license.ini file contains the path to the TopView license file. The default location of the license.ini file is the TopView DataPath (DataPath\license.ini).

If the primary and secondary machines have different TopView license files, they cannot share the license.ini file now located in the shared DataPath location. Therefore, you need to repoint the location of the license.ini file for the primary and secondary machine to a unique location for each machine. You can repoint each machine to the local DataPath folder before it was moved to the shared location (default C:\ProgramData\Exele\TopView\license.ini) or move this local license.ini file to a new location on each machine and set the reference to the new location.

- a. **All TopView versions:** Rename the license.ini file in the new network shared DataPath (to avoid confusion)
- b. **TopView version 7.0 and later:**
Perform the following steps on the primary and secondary machine
 - i. Run the TopView Configurator
 - ii. From the top menu, select Tools > Support apps > Change TopView license location
 - iii. The dialog displays the current location of license.ini which should be the shared DataPath
 - iv. Browse to the license.ini file on the local machine. This should exist in the machine's local DataPath before you changed the DataPath to the share (default C:\ProgramData\Exele\TopView\license.ini)
 - v. Click [OK] to close the dialog
 - vi. Click the [License] button in the upper right corner of the Configurator and verify that the correct TopView license file is specified (primary or failover license file) and that the local license.ini file path is displayed at the bottom of the dialog.

c. **TopView version 6.37 and earlier:**

Perform the following steps on the primary and secondary machine

- i. Locate the existing license.ini file on each machine. This should exist in the machine's local DataPath before you changed the DataPath to the share (default C:\ProgramData\Exele\TopView\license.ini)
- ii. Edit the file %windir%\TopView.ini (usually C:\Windows\TopView.ini)
- iii. Add the line "LicenseFile=*pathtolicenseinifile*" to point to the local license.ini file
 - Example:
Add
LicenseFile=C:\ProgramData\Exele\TopView\License\license.ini
- iv. Save the %windir%\TopView.ini file

7. **OPCUA users:** verify that the certificate information for each machine is pointing to a local folder. See “OPCUA Configuration Files” for more information.
8. Recommended step: move local DataPath files that are no longer required.
Perform this step on the primary and secondary machines.
 - a. Maintaining the local DataPath folders and files can be confusing to someone who is unaware that the DataPath has been repointed to the network share.
 - b. The only files that TopView needs from the local DataPath are
 - i. license.ini
 - ii. Your TopView license file (.txt) if this file also exists in DataPath
If you are unsure of the location of your license file, open lincese.ini in Notepad to view the location
 - iii. OPCUA license files (see previous step)
 - c. Create a subfolder in DataPath named “OLD” and move (not copy) all files from DataPath into the OLD folder *except* the required files mentioned above.

Example: move files and folders from
C:\ProgramData\Exele\TopView\
to
C:\ProgramData\Exele\TopView\OLD
9. Verify the repointed DataPath and license.ini on the primary and secondary machines.
 - a. Run the TopView Configurator
Note: If you are prompted for the TopView license you did not properly repoint your license.ini file. If you are not prompted for the license continue to the next step.
 - b. Select Tools...Open DataPath from the top menu of the Configurator and verify that the network share DataPath location is opened.

10. Set the default location for the configuration files in the TopView Configurator to the new share location.

If you have already used the Configurator to create configuration files, it may be looking at the local DataPath\Config folder as the default location for configuration files

- a. Open the Configurator on the primary and secondary machine
- b. Select "Configuration files" from the left menu
- c. The "Current storage location" displays the default folder for configuration files. If this is pointing to the local machine click the [Set to default] button. The "Current storage location" should now point to the shared DataPath's Config folder
(\\NetworkDevice\TopViewDataPath\Config)

11. Set the configuration file used by each TopView Engine Service to the configuration file in the shared DataPath's Config folder.

If you have already installed one or more TopView Engine Services they were configured to use the configuration file on the local machine. To check the configured location of a TopView Engine Service's configuration file:

- a. Open the Configurator on the primary and secondary machine
- b. Select "TopView Engine Services" from the left menu
- c. In the list of installed Services, check the value of the "Configuration" column. If the location of the configuration file is the local machine:
 - i. Open the configuration file for each Service from the new shared DataPath's Config folder (select "Configuration files" from the left menu and double-click the configuration from the new shared DataPath)
 - ii. Select "Configure Services" from the left menu then click the [Re-install] button to the right of the Service name. You may need to re-enter the configured Logon user password.
 - iii. Verify in the list of TopView Engine Services that the Configuration column now displays the configuration file in the shared DataPath
(\\NetworkDevice\TopViewDataPath\Config)

12. Verify configuration settings for local files.

Some TopView features may specify a local path\file as an input or output of the feature. Verify that these paths now point to the DataPath share location. Common features that may specify local files include:

- a. Snapshot Reports and Snapshot Output (output file)
- b. Alarm Reports (output file)
- c. Audible alarms: WAV files (input)

13. Restart any TopView Engine Services on the primary machine that were previously running

14. Run Admin Tools to verify that the Engines started successfully and are in the “Running” state.

You can view the daily application log for any errors or warnings.

Non-Engine Services

The primary machine may have TopView Services (Windows Services) installed that are not TopView Engines. This includes:

- The TopView Web Configurator
- The Message Receiver (for Microsoft Teams)

The configuration details for these Services are contained in the TopView DataPath. Once the DataPath is shared or copied to the secondary machine, these Services must be installed and started on the secondary machine if they are installed and running on the primary TopView machine.

[Web Configurator](#)

The Web Configurator allows users to make configuration changes through a web browser. If the Web Configurator is enabled on the primary machine, the Service should be installed on the secondary machine.

- There is no method to automatically direct Web Configurator users to the secondary machine
- If a shared DataPath is implemented for TopView Redundancy, the Web Configurator on the secondary machine can be used to make configuration changes.
- If the primary and secondary TopView machines have local DataPaths, and configuration changes are sync'd from the primary to the secondary TopView machine, the Web Configurator on the secondary machine should not be used to make configuration changes.

[Messaging Receiver](#)

The Messaging Receiver Service allows incoming messages for Microsoft Teams. If the Message Receiver Service is enabled on the primary machine, the Service should be installed on the secondary machine.

The configuration of the Message Receiver allows the user to configure both the primary and secondary machine settings (public URL for incoming messages).

WatchDog monitoring: Watchdog tags, Health tags, or Performance Counters?

In TopView Redundancy a WatchDog TopView Engine on the secondary machine monitors the health of TopView Engines on the primary machine. This monitoring can be accomplished with Watchdog tags, health output tags, or Performance Counters.

We recommend using Watchdog tags or Performance Counters for TopView failover/redundancy because they provide direct machine-to-machine monitoring and do not rely upon the data source or individual connections to the data source. If choosing between Watchdog tags or Performance Counters we recommend using Watchdog tags because there are fewer requirements and configuration steps to allow remote monitoring.

If you are planning to implement redundancy with health output tags, please confirm the following:

1. Your TopView data source supports writing to tags
2. TopView has permission to write to a health output tag
3. For TopView PI, if the PI Server implements PI-HA the values written to the health output tag by the primary TopView machine may not be replicated properly to all PI Servers unless the primary machine is configured correctly (n-way buffering). We therefore recommend using Watchdog tags or Performance Counters.

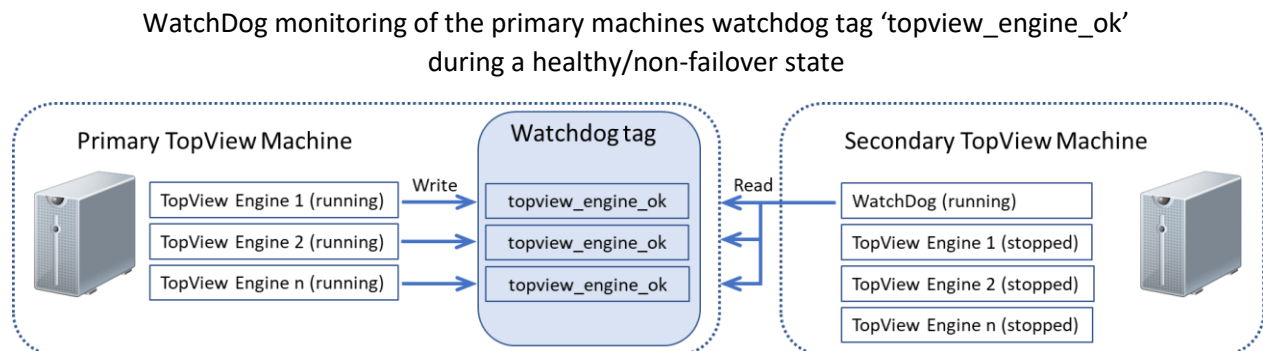
[Watchdog Monitoring: Watchdog tags](#)

Watchdog tags are available in TopView 6.37 and later.

Overview of using Watchdog tags for Watchdog monitoring:

- The TopView data source does not need to support writing to tags
- Each TopView Engine on the primary machine is configured to enable a Watchdog TCP Server on a specified listen port. This will make all Watchdog tags of this primary Engine available to the Watchdog Engine on the secondary machine.
- The WatchDog Engine on the secondary machine monitors the `topview_engine_ok` watchdog tag on the primary machine for each TopView Engine.
- If the `topview_engine_ok` watchdog tag for a TopView Engine on the primary machine indicates that the primary Engine is not healthy, the WatchDog will start a copy of the TopView Engine on the secondary machine (start of failover)
- If the `topview_engine_ok` watchdog tag for a TopView Engine on the primary machine indicates that the primary Engine is healthy, the WatchDog will stop the copy of the TopView Engine on the secondary machine that it had previously started (end of failover)

Note: in addition to the `topview_engine_ok` watchdog tag, the user may decide to implement failover due to other Engine health information available as watchdog tags. This document only provides examples using the `topview_engine_ok` watchdog tag.



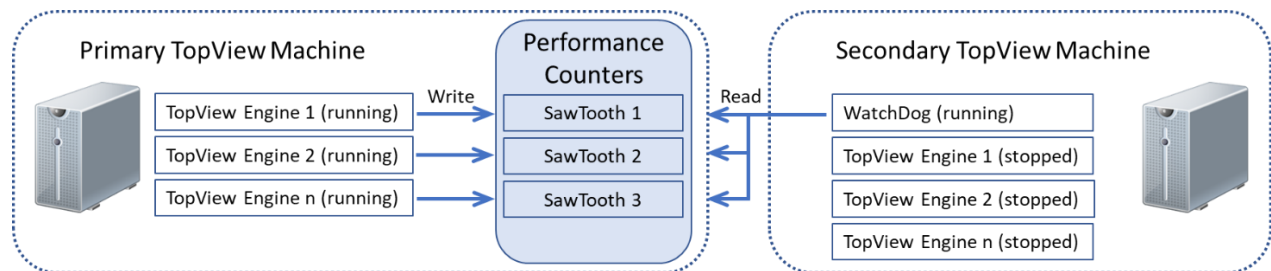
WatchDog Monitoring: Performance Counters

Overview of using [Windows Performance Counters](#) for WatchDog Monitoring:

- The TopView data source does not need to support writing to tags
- Each TopView Engine on the primary machine is configured to create TopView performance counters. One of the TopView performance counters is a sawtooth signal.
 - Note: TopView performance counters are created and written to on the local machine (where the TopView Engine is running).
- The WatchDog Engine on the secondary machine monitors the sawtooth performance counter on the primary machine for each TopView Engine. The WatchDog data source is TopView PerfMon.
- If the sawtooth performance counter for a TopView Engine on the primary machine stops updating for a period of time the WatchDog will start a copy of the TopView Engine on the secondary machine (start of failover)
- If the sawtooth performance counter for a TopView Engine on the primary machine starts updating again the WatchDog will stop the copy of the TopView Engine on the secondary machine that it had started (end of failover)

Note: in addition to the sawtooth performance counter, the user may decide to implement failover due to other health performance counters available in TopView. This document provides examples using the sawtooth performance counter.

WatchDog monitoring of the primary machines performance counters during a healthy/non-failover state

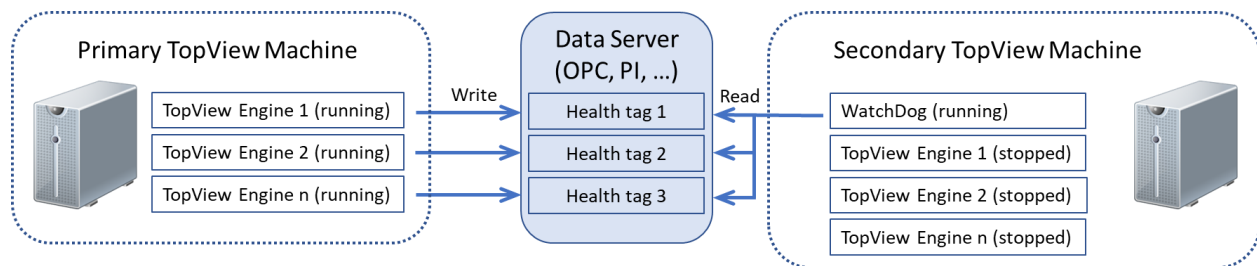


[WatchDog Monitoring: Health tags](#)

Overview of using health tags for WatchDog monitoring:

- The TopView data source must support writing to tags
- Each TopView Engine on the primary machine is configured to output a value to different health tag created on the data server specifically for TopView Redundancy.
 - The value output by each Engine on the primary can be a sawtooth (recommended) or pulse.
 - The health output is configured such that only the primary machine will write to the tag. This allows the secondary machine to use the same TopView Engine configuration file without having the secondary TopView Engine also writing to the health output tag.
- The WatchDog Engine on the secondary machine monitors each health output tag written by the primary TopView Engines. The WatchDog data source is the same as the TopView Engines on the primary machine. Example: if the TopView Engines on the primary machine are TopView OPC, the WatchDog on the secondary machine is also TopView OPC since it is monitoring an OPC tag as the health tag.
- If the health tag stops updating for a period of time the WatchDog will start a copy of the TopView Engine on the secondary machine (start of failover)
- If the health tag starts updating again the WatchDog will stop the copy of the TopView Engine on the secondary machine that it had started (end of failover)

WatchDog monitoring of the primary machine's health tags during a healthy/non-failover state



Configure TopView Redundancy using Watchdog Tags

This section contains details and examples for configuring TopView Redundancy using TopView Watchdog tags (available in TopView 6.37 and later). It assumes that you have read the previous sections including **Watchdog Monitoring: Watchdog tags**.

Primary machine

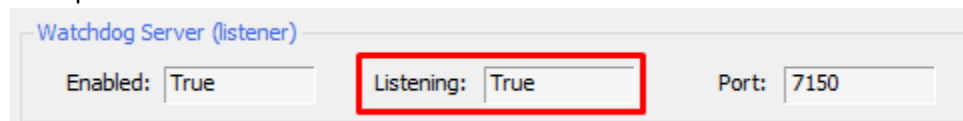
The primary machine is running one or more TopView Engines. Each TopView Engine must be configured to listen for remote connections through its Watchdog TCP Server.

[Configure the Watchdog Server for each Primary Engine](#)

The TopView Engines on the primary machine should be installed to run as a Service with Startup type=Automatic.

For each TopView Engine you will need to perform the following steps:

1. Open the configuration file for the TopView Engine in the TopView Configurator
2. Select “Health” from the left menu
3. Check “Enable Watchdog Server”
 - a. Enter a listen port. The default listen port is 7150.
Warning: If you are configuring redundancy for multiple primary TopView Engines you must enter a different port number for each Engine. The entered port must not be in use by any other Engines or applications on the primary machine.
4. Configure the conditions where topview_engine_ok is not True (v7.1 and later)
5. Save your changes. Start/restart the TopView Engine Service if it is not configured to automatically pick up configuration changes.
6. Verify that the Engine’s Watchdog Server is listening on the configured port:
 - a. Run TopView Admin Tools
 - b. Select the TopView Engine from the dropdown at the top of the screen
 - c. Select “Overview” from the left menu
 - d. In the displayed information, TopView Summary section, verify that listening is True



- e. If it is not listening, select “Application logs” from the left menu and look for any error messages.

[Enable reading Watchdog tags from remote machines \(firewall\)](#)

You will need to read the Watchdog tags on the primary machine from the secondary machine.

On the primary machine:

- Make sure that the Watchdog Server listen port(s) are not being blocked by a firewall
- If possible, turn off the firewall temporarily while you configure the Secondary machine

Secondary machine

The secondary machine will, under normal operation, have the WatchDog TopView Engine running. It will also have copies of the primary TopView Engines installed (as Services) but not running.

The WatchDog Engine will monitor the Watchdog tag `topview_engine_ok` from the primary machine - one for each primary Engine. An alarm condition will be configured for each `topview_engine_ok` tag to detect an unhealthy primary Engine. When the alarm occurs (unhealthy primary Engine) the copy of the Engine will be started on the secondary machine. When the alarm returns to normal (healthy primary Engine) the copy of the Engine will be stopped on the secondary machine.

[TopView license for Watchdog Engine](#)

You do not need a new license for the Watchdog Engine. Watchdog tags are not counted against your TopView license tag count.

[Install TopView Engine Services](#)

Install a copy of each primary TopView Engine Service on the secondary machine.

1. Open the configuration file for each TopView Engine
 - a. Select “Configure Services” from the left menu
 - b. Install the Service using the same LogOn account as the primary machine.
***Make sure you set the “Startup type” to “Manual”**
2. Do not start these TopView Engines
3. Note: The TopView Engine Service name is `topview_cfgrname` where *cfgrname* is the name of the configuration file for the Engine.
Example: for configuration file `myUnitAlarms.cfg` the TopView Service name is `topview_myUnitAlarms`

Create the WatchDog TopView Engine

Create a new configuration

Select File...New configuration from the top menu. Save the configuration as “Watchdog” and ignore any warnings that you do not have any tags/alarms configured.

Configure the Watchdog Server for each primary TopView Engine

1. On the Tags and Limit screen click the [Add other items] button then select “Watchdog tag for another Engine (local or remote)”
Note: this button is at the top of the Tags and Limits screen
2. This will launch the Watchdog Tag Search screen where you can configure Watchdog Servers for each primary Engine and also return Watchdog tags to the TopView monitored tag list.
3. If this is the first primary Watchdog Engine you are configuring it will begin the configuration process, otherwise click the [Add new] button to the right of the “Existing Watchdog Engines” dropdown.
4. Configuration:
 - a. **Computer name or IP address of the TopView machine where the Engine is running:** enter the primary TopView machine name or IP address
 - b. **Enter the name of the configuration (without .cfg) of the Engine you want to monitor:** enter the name of the configuration on the primary machine. This name is only used to create a default name for this new Watchdog Server.
 - c. **Enter the port that the Engine is configured to listen to for watchdog clients:** enter the port number entered on the primary machine for the Engine’s Watchdog Server. See “Configure the Watchdog Server for each Primary Engine” for more details.
 - d. **Enter a unique name to use on this machine when you are monitoring this Engine:** the default name will be the primary engine computer name/IP plus the configuration name.
5. Here is an example Watchdog Server configuration for primary machine named “PrimaryTopView” and primary Engine configuration “MyAlarms”:

Select or configure a TopView Engine to monitor

Existing Watchdog Engines: PrimaryTopView_MyAlarms

TopView machine: PrimaryTopView

Watchdog listen port: 7150
6. Click the [Test connection...] button to test the connection to the primary Engine’s Watchdog Server. Note: the primary Engine must be running for this connection to succeed and the port must not be blocked by a firewall.

7. Repeat these steps for each primary TopView Engine you are configuring for redundancy.

Configure Watchdog tag/alarm for each Primary TopView Engine

For each primary TopView Engine you will create an alarm that looks at the Watchdog tag “topview_engine_ok” for the primary Engine and will start/stop the same Engine on the secondary machine if the primary Engine is not healthy/healthy.

Repeat the following steps for each primary TopView Engine you wish to configure for redundancy:

1. Return to the Watchdog Tag Search: On the Tags and Limit screen click the [Add other items] button then select “Watchdog tag for another Engine (local or remote)”
2. Select the Watchdog Server for the primary Engine from the “Existing Watchdog Engines” dropdown



The image shows a dropdown menu with the text 'Existing Watchdog Engines:' followed by a box containing 'PrimaryTopView_MyAlarms' and a downward arrow icon.

3. In the Available tags list, select tag **topview_engine_ok** and click the [Return selected tags] button
4. The value of Watchdog tag topview_engine_ok will be:
 - a. True if the primary Engine is healthy. By default, “healthy” indicates that we can connect to the Engine’s WatchDog Server and the heartbeat has updated within the last 30 seconds. The details of the healthy condition can be configured in the monitored Engine’s WatchDog Server (the Engine configuration file > Health).
 - b. If the value is not True, the value will be an error message describing the issue. For example: “Can’t connect to Watchdog server” or “heartbeat flatline”
 - c. The status of the value is always good

5. Configure an alarm condition for topview_engine_ok
 - a. Double-click the tag in the monitored tag list to view the Alarm Limits screen
 - b. Set the alarm condition to “not equal to True” to trigger an alarm if the primary TopView Engine is not healthy

Limit #1 ▼

Condition ! ▼ not equal to

Value True Add... ▼

- c. Ensure that the “Check for good status” option on the alarm limit screen is selected
6. Configure the starting and stopping of the same TopView Engine Service on this machine (the secondary machine) upon alarm/rtn of the topview_engine_ok tag
 - a. Select the “Custom Actions” tab
 - b. Under “Start/Stop a TopView Engine Service” select topview_cfgname and “should be running if the current tag/row is ‘In Alarm’”
 - c. Example:

[Refresh list of Engine Services](#)

The TopView Engine Service TopView_MyAlarms ▼ should be RUNNING on this machine

if the current tag/row is ☒ In Alarm ☐ Not in alarm

Otherwise the TopView Engine on this machine should be STOPPED

Install the WatchDog TopView Engine as a Service with Startup type = Automatic

Start and test the WatchDog TopView Engine.

- If you stop the primary Engine, the same Engine on the secondary machine should be started by the Watchdog Engine on the secondary machine within 30 seconds.
- If you start the primary Engine again, the same Engine in the secondary machine should be stopped by the Watchdog Engine once it is able to reconnect to the primary Engine’s Watchdog Server.
- You can use TopView Admin Tools > Overview to see the current values and alarms for topview_engine_ok. Make sure that the Watchdog Engine is selected at the top of the screen.

Configure TopView Redundancy using Performance Counters

This section contains details and examples for configuring TopView Redundancy using Windows Performance Counters. It assumes that you have read the previous sections including **WatchDog Monitoring: Performance Counters**.

Primary machine

The primary machine is running one or more TopView Engines. Each TopView Engine must be configured to write TopView Performance Counters.

[Configure Primary Engines to write Performance Counters](#)

The TopView Engines on the primary machine should be installed to run as a Service with Startup type=Automatic.

For each TopView Engine you will need to perform the following steps:

7. Open the configuration file for the TopView Engine in the TopView Configurator
8. Select “Health” from the left menu
9. Under Performance Counters
 - a. Check “Write to Performance Counters (this configuration)”
 - b. If the [Create] button is enabled, click [Create] to create the performance counters.
Note: the counters should be created during the TopView installation and the [Create] button may not be enabled.
10. Save the configuration
11. Start/restart the TopView Engine Service if it is not configured to automatically pick up configuration changes.
12. Verify that the Engine is writing to the Performance Counters
 - a. Run TopView Admin Tools
 - b. Select the TopView Engine from the dropdown at the top of the screen
 - c. Select “Performance” from the left menu
 - d. Under “TopView Performance Counters” you should see a list of the TopView Performance Counters including one named “heartbeat_5min_sawtooth” which should be updating

[Enable reading Performance Counters from remote machines](#)

You will need to read the Performance Counters on the primary machine from the secondary machine.

On the primary machine:

- Make sure that there is not a firewall running for the required ports. Ports 135 and 445 TCP are used for performance counter monitoring.
- Verify that the Remote Registry and Remote Procedure Call (RPC) Services are running (Control Panel...Administrative Tools...Services)

On the secondary machine there are two users who will need to read Performance Counters on the primary machine:

1. The person configuring the WatchDog TopView Engine in the TopView Configurator
2. The configured LogOn account of the WatchDog TopView Engine Service

If these users are privileged, they should be able to read the Performance Counters on the primary machine. If, while configuring the WatchDog Engine on the secondary machine (next step), you find that these users cannot read the Performance Counters on the primary machine:

1. If the user is a local Windows user account
 - a. The same Windows user name and password must exist on both machines
 - b. The password of the user should not be blank.
2. The user should be a member of 'Administrators', 'Performance Monitor Users', and 'Performance Log Users' groups
3. The user should have read permission to the following registry key on the primary machine:
On the primary machine:
HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg
Right-click winreg, select permissions, and add the users or groups for the two users.
Note: if this key does not exist you may need to create it and then assign permissions.
See [this article](#) for more information.

Workaround if you still cannot access the primary performance counters from the secondary machine you can try the following:

- Edit HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib
 - a. Right click on Perflib key and select permissions
 - b. Click Add and add Local Service with full control
 - c. Save and exit
 - d. Restart the Remote Registry Service or reboot

Secondary machine

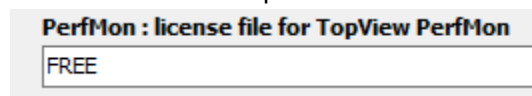
The secondary machine will, under normal operation, have the WatchDog TopView Engine running. It will also have copies of the primary TopView Engines installed but not running.

The WatchDog Engine will monitor the sawtooth Performance Counters from the primary machine - one per primary Engine. An alarm condition will be configured for each monitored Performance Counter to detect an unhealthy primary Engine. When the alarm occurs (unhealthy primary Engine) the copy of the Engine will be started on the secondary machine. When the alarm returns to normal (healthy primary Engine) the copy of the Engine will be stopped.

TopView PerfMon license

Obtain a license for TopView PerfMon:

1. Enable the TopView PerfMon FREE version:
 - i. From the Configurator top menu, select Tools...TopView license
 - ii. In the field for the TopView PerfMon license enter the word FREE



Starting with TopView 6.34, the FREE version of TopView Perfmon will let you monitor any number of TopView performance counters. Before version 6.34 the limit is 5 performance counters.

2. You can also [contact Exele support](#) to obtain a PerfMon license for monitoring TopView Performance Counters.
Once you receive the license file you can reference the file in the TopView PerfMon license file field (from the Configurator top menu, select Tools...TopView license)

Install TopView Engine Services

Install a copy of each primary TopView Engine Service.

4. Open the configuration file for each TopView Engine
 - a. Select "Configure Services" from the left menu
 - b. Install the Service using the same LogOn account as the primary machine.
*Make sure you set the "Startup type" to "Manual"
5. Do not start these TopView Engines
6. Note: The TopView Engine Service name is `topview_cfgname` where *cfgname* is the name of the configuration file for the Engine.
Example: for configuration file `myUnitAlarms.cfg` the TopView Service name is

topview_myUnitAlarms

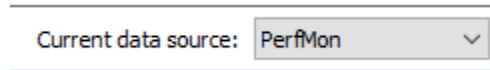
Create the WatchDog TopView Engine

You can manually create the WatchDog Engine configuration or you can use the generation tool.

Manual creation

Note: using “Tool creation” will automatically configure the watchdog. The information provided in this “Manual creation” section can be used to understand the details of the what the tool has created and will allow you to modify the behavior if desired. If you are using TopView 6.34, the tool creation will create the 1-tag solution and you can optionally use this information to implement the 3-tag solution (see “Sawtooth alarm: single tag or 3-tag solution”).

7. Create a new TopView configuration file in the Configurator
 - a. In the upper right corner of the Configurator, select “PerfMon” from the data source dropdown



8. Add the primary sawtooth Performance Counters to the monitored tag list of the WatchDog Engine:
 - a. On the Tags and Limit screen click the [Add tags (tag search)...] button to open the Performance Counter search dialog
 - b. Browse to the name of the primary machine: Click “_Network” in the tree or click the [Add Computer] button and enter the name of the primary machine
 - c. Select the “TopView” category from the listed Performance Counter categories on the primary machine
 - d. Under the category will be a list of running TopView Engines.
For each TopView Engine:
 - i. Select the TopView Engine name
 - ii. Select the counter named “heartbeat_5min_sawtooth” and click the [Add] button
 - e. Click [Return selected counters] to return the counters to the monitored tag list

Sawtooth alarm: single tag or 3-tag solution

The TopView Watchdog will monitor the sawtooth counter of the primary Engine and will create an alarm condition if the primary Engine is not healthy. The alarm condition will look for the sawtooth counter not changing value (primary engine is not updating the counter) or if the sawtooth counter has bad status (primary engine sawtooth counter cannot be read).

There are two options for how you configure the alarm

- Single tag: configure 2 alarm conditions of one monitored heartbeat_5min sawtooth tag
 - *This is no longer the recommended solution
 - a. Configure alarm conditions “flatline for 30 seconds” or “bad status”
 - b. The sawtooth counter alarm will control starting/stopping the failover Engine
 - c. Advantages: easy to configure in one sawtooth tag
 - d. Disadvantages: possible multiple alarm conditions (and start/stop of the failover Engine) during failover since the two alarm conditions are not mutually exclusive.
- 3-tag: Monitor the two alarm conditions separately (the heartbeat_5min sawtooth counter tag is added twice to the Watchdog monitored tag list) and a 3rd “user tag” that monitors the alarm state of the two sawtooth alarms
 - *This is now the recommended solution
 - a. Configure the alarm condition of the first sawtooth counter tag as “flatline for 30 seconds”
 - b. Configure the alarm condition of the second sawtooth counter tag as “bad status”
 - c. Configure a “user tag” that will be in alarm if the first or second sawtooth tag is in alarm. The user tag “in alarm” state will control starting/stopping the failover Engine.
 - d. Advantages:
 - i. tracks the logs the sawtooth flatline and bad status alarms separately
 - ii. will not create multiple alarm conditions (of the user tag) during a failover, therefore only one start/stop event of the failover Engine during a failover event
 - e. Disadvantages:
 - i. More configuration steps (if not using the Watchdog creation tool)
 - f. Note: the 3-tag solution is used by the Watchdog creation tool

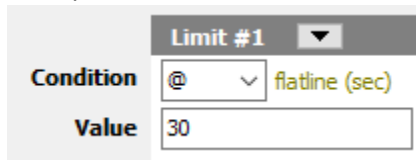
Single-tag solution

This option monitors a single sawtooth counter tag for each primary Engine that you are configuring for failover. This is also the result of the Watchdog creation tool in version 6.34 although we now recommend the 3-tag solution. See “Sawtooth alarm: single tag or 3-tag solution” for an overview of the two solution options.

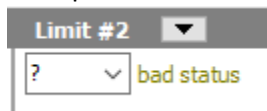
Configure the alarm conditions of the heartbeat_5min sawtooth tag.

For each heartbeat_5min_sawtooth tag (one per primary TopView Engine):

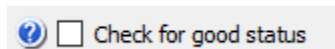
- Configure an alarm condition to detect that the counter is not updating:
Alarm limit #1: value flatline for X seconds
condition: @ (value flatline)
value: X (for X seconds)
The counter should be updating at the refresh rate of the primary TopView Engine (default=5sec) so a reasonable flatline alarm is 30 seconds
Example:



- Configure a second alarm condition to detect that the counter is not available. This indicates that the primary machine is unavailable or that the primary Engine is not running:
Alarm limit #2: bad status alarm
condition: ? (bad status)
Example:



Ensure that the “Check for good status” option on the alarm limit screen is NOT selected



- Configure the starting and stopping of the TopView Engine Service on the secondary machine upon alarm/rtn of the sawtooth Performance Counter.
- In TopView 6.35 and later
 - Select the “Custom Actions” tab
 - Under “Start/Stop a TopView Engine Service” select `topview_cfgname` and “should be running if the current tag/row is ‘In Alarm’ “
 - Example

The TopView Engine Service `TopView_myUnitAlarms` should be RUNNING on this machine

if the current tag/row is ☒ In Alarm ☐ Not in alarm

Otherwise the TopView Engine on this machine should be STOPPED

- In TopView 6.34 and earlier
 - Select the “Custom Response” tab
 - For the into alarm run string set
Run string = net
Command line = **start** `topview_cfgname`
 - For the return-to-normal run string set
Run string = net
Command line = **stop** `topview_cfgname`

Example:

Settings

This tag/row into alarm: Run string (exe, bat, ...) `net` Command line argument(s) `start topview_myUnitAlarms` Test

This tag/row return to normal: Run string (exe, bat, ...) `net` Command line argument(s) `stop topview_myUnitAlarms` Test

9. Install the WatchDog TopView Engine as a Service with Startup type = Automatic
10. Start and test the WatchDog TopView Engine

3-tag solution

This option monitors 2 instances of the sawtooth counter tag plus a “user tag” for each primary Engine that you are configuring for failover. This is the result of the Watchdog creation tool in version 6.35 and later. See “Sawtooth alarm: single tag or 3-tag solution” for an overview of the two solution options.

At this point you should have a single sawtooth counter tag in the monitored tag list. We need to add two more tags

- Right-click the sawtooth counter tag in the monitored tag list and select “Duplicate tag (insert after selected tag)”. This should add a copy of the sawtooth counter tag.
- Click the [Add other items...] button, then “User tag”
 - When prompted for the custom name, enter `Failover_cfgname`
where `cfgname` is the name of the primary TopView engine configuration

- Example:

Primary TopView machine name is PrimaryMachineName

Primary Engine configuration is myUnitAlarms.cfg

> Row	Limits	Tag	Server
1		TopView_.heartbeat_5min_sawtooth_.myUnitAlarms	PrimaryMachineName
2		TopView_.heartbeat_5min_sawtooth_.myUnitAlarms	PrimaryMachineName
3		user_tag_Failover_myUnitAlarms	<Local computer>

Configure the alarm condition of the first heartbeat_5min sawtooth tag

- Select the first heartbeat tag in the monitored list
- To the right of the tag list on the General tab, enter the RowUID as `cfgname_heartbeat_value`
where `cfgname` is the name of the primary TopView Engine configuration
Example: `myUnitAlarms_heartbeat_value`
- Configure one alarm condition to detect that the counter is not updating:
Alarm limit #1: value flatline for X seconds
condition: @ (value flatline)
value: X (for X seconds)
The counter should be updating at the refresh rate of the primary TopView Engine (default=5sec) so a reasonable flatline alarm is 30 seconds
Example:

	Limit #1
Condition	@ flatline (sec)
Value	30

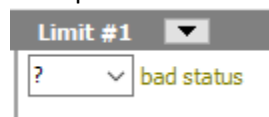
Configure the alarm condition of the second heartbeat_5min sawtooth tag

- Select the second heartbeat tag in the monitored list
- To the right of the tag list on the General tab, enter the RowUID as *cfgname_heartbeat_status* where cfgname is the name of the primary TopView Engine configuration
Example: myUnitAlarms_heartbeat_status
- Configure one alarm condition to detect that the counter is not available. This indicates that the primary machine is unavailable or that the primary Engine is not running:

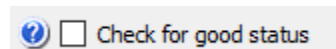
Alarm limit #1: bad status alarm

condition: ? (bad status)

Example:



Ensure that the “Check for good status” option on the alarm limit screen is NOT selected



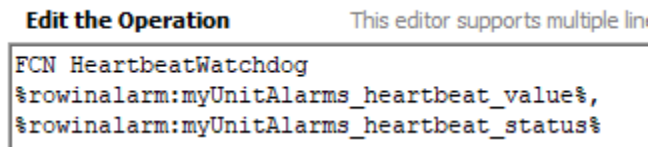
Configure the alarm condition of the user tag

- The user tag value will be the result of a logic function that looks at the alarm state of the two sawtooth tags (flatline, bad status) and returns 1 if either is in alarm, 0 otherwise. The function's reference to each of the sawtooth tag will use the RowUID entered for each sawtooth tag.
 - Select the user tag in the monitored tag list
 - To the right of the tag list on the Operations tab, click [Operation Editor] and enter the Operation as:

```
FCN HeartbeatWatchdog
%rowinalarm: cfgname _heartbeat_value%,
%rowinalarm: cfgname _heartbeat_status%
```

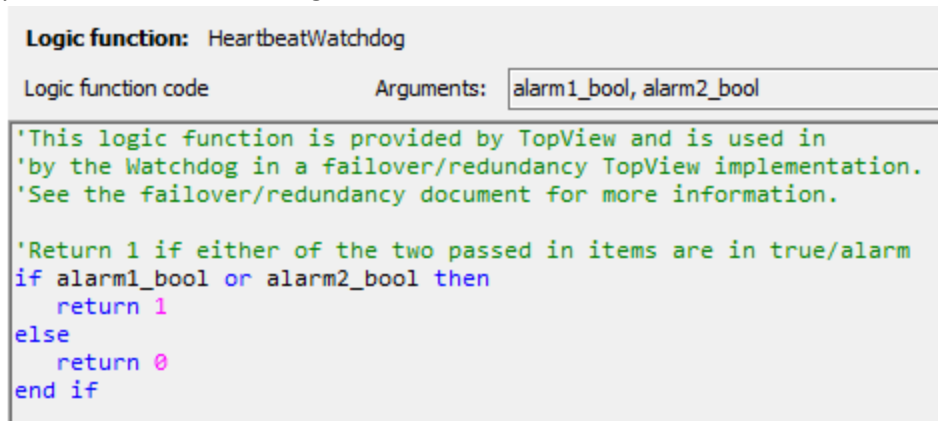
where *cfgname* is the name of the primary TopView Engine configuration.

- Example:



```
Edit the Operation          This editor supports multiple lines
FCN HeartbeatWatchdog
%rowinalarm:myUnitAlarms_heartbeat_value%,
%rowinalarm:myUnitAlarms_heartbeat_status%
```

- The Logic Function “HeartbeatWatchdog” should exist for TopView v6.35 and later. If you need to create this Logic Function the details are



```
Logic function: HeartbeatWatchdog
Logic function code      Arguments: alarm1_bool, alarm2_bool

'This logic function is provided by TopView and is used in
'by the Watchdog in a failover/redundancy TopView implementation.
'See the failover/redundancy document for more information.

'Return 1 if either of the two passed in items are in true/alarm
if alarm1_bool or alarm2_bool then
    return 1
else
    return 0
end if
```

- Configure one alarm condition of the user tag to detect when the user tag value (the logic function result) is equal to 1. This indicates that one or both of the sawtooth tags are in alarm.:

Alarm limit #1: value is equal to 1

condition: = (equal to)

value: 1

Example:

	Limit #1 ▼
Condition	= ▼ equal to
Value	1

- For the user tag, configure the starting and stopping of the TopView Engine Service on the secondary machine upon alarm/rtn of the sawtooth Performance Counter.
 - In TopView 6.35 and later
 - Select the “Custom Actions” tab
 - Under “Start/Stop a TopView Engine Service” select `topview_cfgrname` and “should be running if the current tag/row is ‘In Alarm’ “
 - Example

The TopView Engine Service [Refresh list of Engine Services](#) `topview_myUnitAlarms` should be RUNNING on this machine

if the current tag/row is ☒ In Alarm ☐ Not in alarm

Otherwise the TopView Engine on this machine should be STOPPED

- In TopView 6.34 and earlier
 - Select the “Custom Response” tab
 - For the into alarm run string set
Run string = net
Command line = **start** `topview_cfgrname`
 - For the return-to-normal run string set
Run string = net
Command line = **stop** `topview_cfgrname`

Example:

Settings

	Run string (exe, bat, ...)	Command line argument(s)	
This tag/row into alarm:	net	start topview_myUnitAlarms	Test
This tag/row return to normal:	net	stop topview_myUnitAlarms	Test

11. Install the WatchDog TopView Engine as a Service with Startup type = Automatic

12. Start and test the WatchDog TopView Engine

Tool creation

Starting with TopView 6.34, the WatchDog configuration tool can be used to create the WatchDog Engine configuration based on the details provided in “Manual creation”. In version 6.34, the configuration result of this tool is the 1-tag solution. In version 6.35 and later the configuration result is the recommended 3-tag solution. See “Sawtooth alarm: single tag or 3-tag solution” for an overview of the two solution options.

- Open the TopView Configurator on the secondary machine.
- From the top menu, select Tools...Create WatchDog configuration

Configure the details of the WatchDog:

1. Enter a name for the WatchDog Engine configuration file
2. Enter the name of the primary TopView computer
3. Select the primary TopView Engine(s) to monitor.
The list of Engine installed TopView Engines is displayed. Check the name of each Engine that you are configuring for failover/redundancy. This should be the list of Engines that you installed on the secondary machine in “Install TopView Engine Services”.
4. Failover condition:
 - a. Heartbeat counter (recommended): this will configure failover to occur when the primary Engine’s heartbeat performance counter is not updating. This is the same configuration explained in “Manual creation”.
 - b. Counters to monitor: select one or more counters to monitor for each primary Engine. You will need to manually configure the alarm condition for each of these counters.
5. WatchDog Service
 - a. Enable if you would like the tool to install the WatchDog Engine Service. The entered Service LogOn user must have permission to read performance counters on the primary machine.

Once you have generated the WatchDog configuration you can close the tool and open the WatchDog configuration file. If you selected “Failover condition: Heartbeat counter” you can verify the configured settings against those explained in “Manual creation”. If you selected “Failover condition: Counters to monitor” you will need to configure the conditions and custom response for each counter similar to the heartbeat counter configuration.

Start and test the WatchDog TopView Engine.

Configure TopView Redundancy using Health Output Tags

This section contains details and examples for configuring TopView Redundancy using health output tags. It assumes that you have read the previous sections including **WatchDog Monitoring: Health tags**

Primary machine

The primary machine is running one or more TopView Engines.

Each TopView Engine must be configured to write to a different health output tag. You will need to create these tags in your data server (OPC Server, PI Server, ...) and ensure that the TopView Engines have write permission to these tags.

[Configure Primary Engines to write to health output tags](#)

The TopView Engines on the primary machine should be installed to run as a Service with Startup type=Automatic.

For each TopView Engine you will need to perform the following steps:

1. Open the configuration file for the TopView Engine in the TopView Configurator
2. Select "Output Points" from the left menu
3. Create a health Output Point:
 - a. Click [Add new Output Point] and enter a name for the output point
 - b. Output tag: the name of the health output tag. If your data source supports browsing you can click the [Tag Search...] button to select the tag
 - c. Once the output tag name is entered you can test writing a value to the tag using the [Test output] button
 - d. Output signal type: choose Health output: Sawtooth
 - e. Example:

The screenshot shows two windows from the TopView Configurator. The top window, titled 'Existing Output Points', contains a table with the following data:

Name	Output tag (server tag)	Signal type
HealthOutputPoint	MyServer OUT.Float.OTAGF0001	Health

To the right of the table are buttons: '+ Add new Output Point', '- Remove Output Point', and a 'Bulk Configuration' section with 'Export' and 'Import' buttons. The bottom window, titled 'Selected Output point', shows the configuration for 'HealthOutputPoint'. It includes fields for 'Name' (HealthOutputPoint) and 'Output Tag' (||MyServer||OUT.Float.OTAGF0001), with 'Tag Search...', 'Test output', and 'Tag properties...' buttons. Under 'Output signal type', the 'Health output: Sawtooth' option is selected. It specifies a cycle every 5 minutes from 0 to 100. There is also an option for 'Event output' with a 'Check' button and a checkbox for 'On startup, output value'.

- f. Click [OK]
4. Configure the TopView Engine to write to the health Output Point
 - a. Select “Health” from the left menu
 - b. Under Heartbeat settings, check “Enable heartbeat signal output”
 - c. For Heartbeat signal #1, use the dropdown to select the health output point you created
 - d. Check “only output heartbeat#1 if the TopView Engine is running on computer named:”
 - e. Click [Get this computer name] to fill the computer name field with the name of the primary machine

f. Example:

☒ Enable heartbeat signal output

You can configure one or two heartbeat signals. Use two signals if you would like to (1) output a heartbeat signal to two different Servers or (2) have separate signals for the primary and secondary TopView instance when using TopView failover.

Output Points... Each heartbeat signal is defined by creating a Health Output Point. The Output Point defines the server, output tag, and signal type. Once the Health Output Point is defined, you can assign it to one of the two heartbeat signals below.

You should configure different heartbeat output tags for each TopView configuration.

Select a health output point for each heartbeat signal

Heartbeat signal #1: HealthOutputPoint

☒ Only output heartbeat #1 if the TopView Engine is running on computer named: TopViewPrimary

Typically the primary computer name in a TopView failover configuration

Get this computer name

5. Save the configuration
6. Start/restart the TopView Engine Service if it is not configured to automatically pick up configuration changes.
7. Verify that the Engine is writing to the heartbeat tag
 - a. Select “Output Points” from the left menu
 - b. Select the health Output Point from the list
 - c. Click Tag Properties to view the current value of the tag. Perform this step a few times to make sure that the value is changing
 - d. If the health output tag is not changing
 - i. Use TopView Admin Tools to view the application log for errors.
 - ii. Most likely issue: the TopView Engine does not have permission to write to the tag. If you are running the TopView Engine as a Service you need to make sure that the configured LogOn account of the Service is allowed to write to the tag.

Secondary machine

The secondary machine will, under normal operation, have the WatchDog TopView Engine running. It will also have copies of the primary TopView Engines installed but not running.

The WatchDog Engine will monitor each health tag from the primary machine - one per primary Engine. An alarm condition will be configured for each monitored health tag to detect an unhealthy primary Engine. When the alarm occurs (unhealthy primary Engine) the copy of the Engine will be started on the secondary machine. When the alarm returns to normal (healthy primary Engine) the copy of the Engine will be stopped.

Perform the following steps on the secondary machine:

- Install a copy of each primary TopView Engine Service.
 - a. Open the configuration file for each TopView Engine
 - Select “Configure Services” from the left menu
 - Install the Service using the same LogOn account as the primary machine.
*Make sure you set the “Startup type” to “Manual”
 - b. Do not start these TopView Engines
 - c. Note: The TopView Engine Service name is `topview_cfgname` where *cfgname* is the name of the configuration file for the Engine.
Example: for configuration file `myUnitAlarms.cfg` the TopView Service name is `topview_myUnitAlarms`
- Create the WatchDog TopView Engine
 - a. Create a new TopView configuration file in the Configurator
 - b. Add the health tags to the monitored tag list (one per primary TopView Engine)
- Configure the WatchDog TopView Engine
For each health tag:
 - a. Configure an alarm condition to detect that the health tag is not updating:
Alarm limit#1: value flatline for X seconds
condition = @ (value flatline)
value = X (for X seconds)
Each health tag should be updating at the refresh rate of the primary TopView Engine (default=5sec) so a reasonable flatline alarm is 30 seconds
Example:

The screenshot shows a configuration window for 'Limit #1'. It has two main sections: 'Condition' and 'Value'. The 'Condition' section has a dropdown menu with '@' selected and a text field with 'flatline (sec)'. The 'Value' section has a text field with '30'.

Limit #1	
Condition	@ flatline (sec)
Value	30

- b. Configure the starting and stopping of the TopView Engine Service on the secondary machine upon alarm/rtn of the health tag.

- TopView 6.35 and later

- Select the “Custom Actions” tab
- Under “Start/Stop a TopView Engine Service” select topview_cfgname and “should be running if the current tag/row is ‘In Alarm’ ”

- Example

The screenshot shows a configuration window for a TopView Engine Service. At the top right is a link "Refresh list of Engine Services". Below it, a dropdown menu is set to "TopView_myUnitAlarms". To the right of the dropdown, it says "should be RUNNING on this machine". Below this, there are two radio buttons: "In Alarm" (which is selected) and "Not in alarm". At the bottom, it says "Otherwise the TopView Engine on this machine should be STOPPED".

- TopView 6.34 and earlier

- Select the “Custom Response” tab
- For the into alarm run string set
Run string = net
Command line = **start** topview_cfgname
- For the return-to-normal run string set
Run string = net
Command line = **stop** topview_cfgname

- Example:

The screenshot shows a "Settings" window with two sections. The first section is "This tag/row into alarm:" and the second is "This tag/row return to normal:". Each section has two input fields: "Run string (exe, bat, ...)" and "Command line argument(s)". For the "into alarm" section, the run string is "net" and the command line argument is "start topview_myUnitAlarms". For the "return to normal" section, the run string is "net" and the command line argument is "stop topview_myUnitAlarms". Each input field has a "Test" button next to it.

- Save the WatchDog configuration
- Install the WatchDog TopView Engine as a Service with Startup type = Automatic
- Start the WatchDog TopView Engine

Additional Failover/Redundancy components

VOIP Callout Failover – Grandstream IP PBX

TopView's Voice Notification supports VOIP-based phone calls. [One of our suggested solutions](#) to use VOIP with an analog line is to install an IP/PBX which contains a VOIP SIP Server.

Part of the TopView configuration is specifying the SIP Server in the IP PBX by entering the IP address or host name of the IP PBX.

In a TopView Redundancy configuration, the primary and secondary machines may share a single IP PBX/SIP or they may each have their own dedicated IP PBX/SIP.

[One shared Grandstream device](#)

If the primary and secondary machines are using one IP PBX/SIP, the shared TopView configuration file can specify the single SIP Server by IP address.

[Two different Grandstream devices](#)

If the primary and secondary machines each have their own IP PBX/SIP, the shared TopView configuration cannot specify the SIP Server by IP address since each machine needs to use a different IP PBX/SIP.

The solution is to create a host name for the IP PBX on the primary and secondary machines that specify the correct IP address of the IP PBX/SIP. This locally-defined host name can then be used as the SIP Server name in the shared TopView configuration file.

To set the host name and IP address on the primary and secondary machine, you must edit the hosts file located on each machine and create an entry for the SIP Server with the correct IP address.

The default location for the host file is: C:\Windows\System32\drivers\etc\hosts

Example host file:

```
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host
# name.
# The IP address and the host name should be separated by at least one
# space.
192.168.1.182    MySIPServer
```

OPCUA Configuration Files

The following information is for TopView OPC/SCADA users implementing the OPCUA data source.

Overview

OPCUA configuration information is stored in the DataPath\Config\OpcUa folder.

The OPCUA configuration information includes

- Defined OPCUA Server Aliases (OpcUaAliases.json)
- TopView OPCUA behavior and configuration settings (OpcUaConfig.json)
 - a. This file contains the path to the certificates
- OPCUA certificates (pki folder)

How OPC Server Aliases are resolved

- TopView reads the OPCUA Server Alias settings defined in DataPath\Config\OpcUa\OpcUaAliases.json

How certificates are located

- TopView reads the settings file (DataPath\Config\OpcUa\OpcUaConfig.json)
- From this file, TopView retrieves the path to the certificates folder (json value CertificatesStorePath)
- TopView uses this path as the root folder for OPCUA certificates

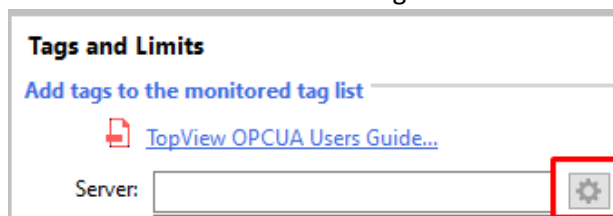
CertificatesStorePath: When TopView is first configured, the current local path to the certificates folder is stored to CertificatesStorePath in OpcUaConfig.json

Default: C:\ProgramData\Exele\TopView\Config\OpcUa\pki

If you move the TopView DataPath, CertificatesStorePath will still point to the original location.

Viewing/Changing CertificatesStorePath:

- Launch the TopView Configurator
- Go into the OPCUA Server settings



- Click the [UA Settings button]
- Change the "Certificate store location"

[Sharing OPCUA information](#)

Although many of the OPCUA settings can be shared between the primary and secondary TopView machine, there are some settings that are machine-specific (e.g., own certificate). Therefore, we recommend that each TopView machine be configured to use a local folder for the storage of certificate information, and that this information is not shared between the two machines.

[Local Datapath](#)

If you have configured TopView redundancy such that the primary and secondary machine each have their own local DataPath:

- Configure and verify connection to the OPCUA Server on each machine
 - This will update the settings in the local DataPath\Config\OpcUa folder
 - Use the same OPCUA Server Alias name on each machine. The OPCUA Server Alias name is used by TopView configurations as the “server” for each OPCUA tag. The resolution of this alias to an OPCUA Server is performed using the settings in the local DataPath\Config\OpcUa folder.
- When syncing the primary DataPath to the secondary DataPath, **do not sync** the Config\OpcUa folder.

[Shared DataPath](#)

If you have configured TopView redundancy such the primary and secondary machine are using a shared DataPath:

- Make sure that each machine is looking at a local folder for OPCUA certificates. See CertificatesStorePath in “How certificates are located” to view/edit this path.
- Configure and verify connection to the OPCUA Server on each machine
 - Both machines are sharing the OPCUA Server alias settings in the shared DataPath\Config\OpcUa\OpcUaAliases.json file
 - The certificates for each machine are stored in the local CertificatesStorePath folder specified in the shared DataPath\Config\OpcUa\OpcUaConfig.json file which should be pointing to a local folder.