



# Configuring Auto-Responses

© 3Ds (UK) Limited, October, 2013  
<http://www.Sentry-go.com>

*Be Proactive, Not Reactive!*

A typical monitoring solution will monitor your software or environment and, in the event an issue is detected, let you know. It may allow you to be alerted in a number of ways, depending on the fault and even the day or time the fault occurs. Some, like the Sentry-go Monitoring System go further by giving you the option of allowing the monitor itself to attempt to resolve the error itself by performing an automated action or response. This can be an extremely powerful feature and is certainly a subject worthy of more than a quick glance.

## In this guide

Why use auto-responses ? .....	2
Configuring an automatic response .....	3
Auto-response considerations .....	10
Consider All the Options .....	12
Some Dos & Don'ts .....	12
Local vs Remote Responses .....	15
More Information .....	19

## Why use auto-responses ?

There are a number of reasons why you might consider configuring automatic responses ...

- Some faults can actually occur quite frequently. They're more annoying than anything else, but require time and effort to resolve, *each & every time they occur!*
- Even with the quickest forms of alert, a manual response will be delayed. You may have to access a machine & logon, you may also be involved with another issue, or simply be on the phone and not see the message for a few minutes.
- For important issues or systems, every second counts. The faster the problem is resolved, the less people (or other systems) are affected - which is ultimately better for them, you and your business.

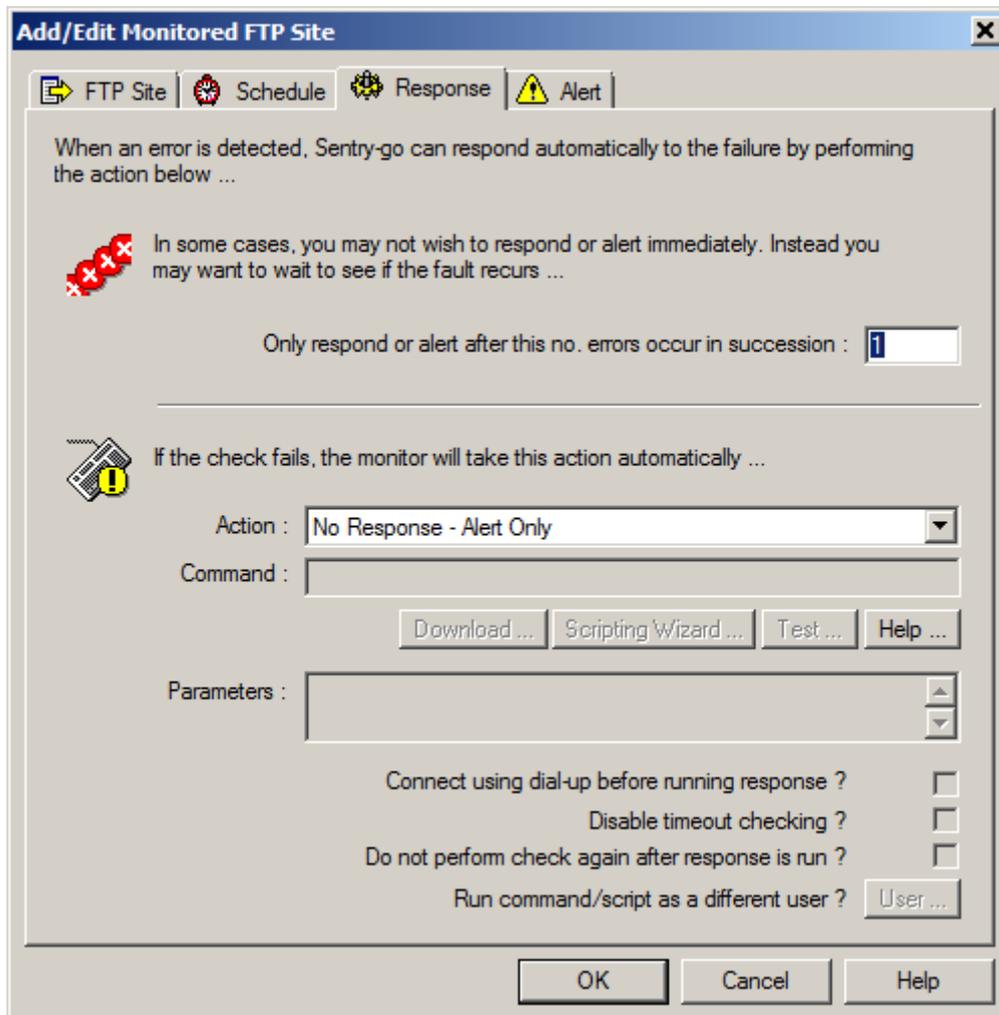
Of course, we're not saying everything can, or should be automated and we're certainly not saying the Administrators shouldn't be kept informed. But take the following examples ...

- A key service stops running on your server
- Your server runs low on disk space preventing prints from being submitted or data being written to the database
- Someone runs Setup on your live server without the Administrators knowledge
- A database table becomes locked such that your orders cannot be accepted
- Your web server returns error pages rather than your shiny home page

In all of these, and many more, you could easily take advantage of an automated approach, at the very least to get the system back & running in the quickest possible time, and investigate further thereafter.

## Configuring an automatic response

Automatic responses are configured as part of an individual check. To create a response, simply configure the check and select the "Response" tab ...



By default, no automatic action will be taken and the failure will trigger an alert. However, you can define any automatic response from here as described below.

### Only alert after this no. errors occur in succession

If you only want to take automatic action (or alert) after the check has failed continually for a given number of times, enter this number in the first option. The default value is 1, meaning only a single error has to be detected before action is taken.

 This allows for conditions such as "High CPU" monitoring where one-off errors can be accepted but continual faults to be detected and optionally resolved.

The count is automatically reset once the test has run successfully.

## Action

This selection list defines what you want to do in response to the check failing. By default, no action is taken and the alert triggered.

 The options presented here are dependent on the check being configured. Some of those listed below are only shown for certain types of monitoring check.

Selected Action	Meaning
<b>No Response - Notify Only</b>	Select this option if you simply wish to alert one or more System Administrators about the error. In this case, no corrective action will be attempted by the monitor.
<b>No Response - Log Only</b>	If logging is enabled via the "Logging" tab, either to file, database or both, then details of all alerts will be written to the appropriate location.  If this option is set, no notification or action will be triggered. Instead, the alert will simply be written to the log.
<b>Reboot the Server</b>	Select this option if the monitor should reboot the server in response to this failure. In this case, no other corrective action will be attempted.  <i>Before a reboot occurs, notifications, indicating that a reboot is about to occur will be automatically sent to the appropriate System Administrators based on the Alert group of the error. A short delay will precede any reboot, to ensure the above Administrators receive their notification.</i>   If, after the server is rebooted, the same alert is triggered again, the same Administrators will be notified but the reboot cancelled. This prevents continual reboots from being performed.
<b>Run an Application</b>	This option allows the monitor to take automatic corrective action to the appropriate test by running the application entered in the "With this Command" field.  For example, to clear down temporary files to reclaim free disk space etc. you might run an application or batch file.   The application entered should be specified along with the full path & must be available to the monitor being configured.  The application is run directly from the location entered. No changes or modifications are made by the monitor to this file.

## Run a Command

This option allows the monitor to take automatic corrective action to the appropriate test.

If selected, the "With this Command" option is enabled, allowing you to specify the command you wish to run in response to the failure.



The command entered should be specified - along with the full path if required & must be available to the monitor being configured.

The monitor will run the command using the Windows "CMD" interface directly from the location entered. No changes or modifications are made by the monitor to this file.

## Run a VBScript

This option also allows the monitor to take automatic corrective action to the appropriate test.

If selected, the "With this Command" option is enabled, allowing you to specify a Windows Scripting Host (WSH) or VBScript (VBS) file which should be run in response to the failure.



The script entered should be specified along with the full path & must be available to the monitor being configured.

In addition, place-markers can be used within the script to include context specific information relating to the fault that triggered the response. These will be translated at runtime, prior to running the response.

For more information see the "Sentry-go Place-markers" guide.

Example scripts can be found on-line at <http://www.sentry-go.com/sentry-go-scripting-library.aspx>.

## Restart the Service

Select this option to attempt to restart the associated service which has been detected as stopped.

## Restart or Continue the Service

Select this option to attempt to restart the associated service which has been detected as stopped. In addition, the monitor will respond to a "Paused" state by attempting to continue the service.

## Start the Process with this Command

Select this option to attempt to run the required number of copies of the process with the command specified below in order to meet the specified check criteria.

## Attempt to Terminate the Process

Select this option to terminate the running copies of the given application if they are found to be running.



It is assumed that the process running has no user session and will be terminated without warning. Any user or associated data will be lost.

### **Pause the Printer**

If this option is selected, the associated printer will automatically be paused, stopping any further documents from being printed. Documents will continue to be queued.



The printer may fail to pause if a local printer error is being reported.

### **Resume the Printer**

If this option is selected, Sentry-go will attempt to resume (continue) the printing of documents from a previously paused state.



The printer may fail to resume if a local printer error is being reported.

### **Delete all Queued Documents**

If this option is selected, Sentry-go will purge (delete) all documents from the associated printer/print queue.

### **Take the Printer Off-line**

If this option is selected, Sentry-go will attempt to take the printer off-line (i.e. jobs will be spooled and held - not printed).



This may have no noticeable effect if an error is being reported by the printer.

### **Bring the Printer On-line**

If this option is selected, Sentry-go will attempt to bring the printer on-line (i.e. documents will be printed, not simply spooled and held).



This may have no noticeable effect if an error is being reported by the printer.

### **Redirect Print Queue**

If this option is selected, Sentry-go will attempt to redirect the print queue to the local port entered below & bring the printer on-line (i.e. documents will be printed, not simply spooled and held).



The target port must point to the same type of printer, as the spooled data will have already been converted for that printer type.

The document currently printing (i.e. the document effectively reporting the error) cannot be transferred to another printer.

If the associated printer is offline or is showing an error, the document showing that error cannot be moved to a new port.

### **Pause Queued Document**

If this option is selected, the associated document will be paused (not printed).



In most cases, if the document has started printing, it will complete, even if it is paused.

The response has no effect for printer-wide checks - i.e. checks not linked to print jobs.

### **Delete all Queued Documents**

If this option is selected, Sentry-go will attempt to delete all queued documents from the associated print queue.

<b>Resume Queued Document</b>	<p>If this option is selected, the associated document will be un-paused (i.e. resumed).</p> <p> The response has no effect for printer-wide checks – i.e. checks not linked to print jobs.</p>
<b>Restart Document Printing</b>	<p>If this option is selected, the printing of the associated document will be restarted from the beginning.</p> <p> The response has no effect for printer-wide checks – i.e. checks not linked to print jobs.</p>
<b>Delete the Queued Document</b>	<p>If this option is selected, Sentry-go will attempt to delete the associated document.</p> <p> The response has no effect for printer-wide checks – i.e. checks not linked to print jobs.</p>
<b>Terminate the blocking process</b>	<p>If this option is selected, Sentry-go will attempt to cancel the SQL connection associated with the connection that is stopping another SQL process(es) from continuing – i.e. the blocking SQL process.</p> <p> No warning will be given &amp; any unsaved (uncommitted) data associated with the connection will be lost. However, blocked processes will then be allowed to continue.</p>

***The following options automatically invoke the Sentry-go Scripting Wizard to generate the appropriate logic. No scripting knowledge is required in order to do this, though you can edit or further enhance the generated script if you wish.***

***Depending on the check originally performed, some of these options may not be available.***

<b>Clear down temporary files to reclaim disk space</b>	Select this option to automatically delete temporary files in order to reclaim disk space, in response to a triggered alert.
<b>Stop/start the print spooler</b>	Select this option to stop/start or restart the Windows Print Spooler service, in response to a triggered alert.
<b>Stop/start the IIS Web service</b>	Select this option to stop/start or restart the Internet Information Services (IIS) web service, in response to a triggered alert.
<b>Stop/start the named Windows service</b>	Select this option to stop/start or restart a named Windows service, in response to a triggered alert.
<b>Run an SQL Query or Command</b>	Select this option to connect to a SQL Server & run SQL or database commands, optionally verifying results, in response to a triggered alert.
<b>Dismount a disk drive</b>	Select this option to dismount the selected disk drive, in response to a triggered alert.
<b>Restart all COM+ applications</b>	Select this option to stop & restart all configured COM+ applications, in response to a triggered alert.

<b>Stop a named COM+ application</b>	Select this option to stop a specific COM+ application, in response to a triggered alert. The application will typically restart after this automatically when requested by the caller.
<b>Stop all named COM+ applications</b>	Select this option to terminate all COM+ applications, in response to a triggered alert. Applications will typically restart after this automatically when requested by the caller.
<b>Unmap a network drive</b>	Select this option to automatically un-map a network drive, in response to a triggered alert.
<b>Remap network drives to a new server</b>	Select this option to automatically re-map similarly named drives from one server to a new one, in response to a triggered alert.  This is useful if you are running a mirrored or back-up server and wish to reference this second server in response to a failure of the first.
<b>Reboot a remote server</b>	Select this option if you want to reboot (restart) a named remote server.
<b>Shutdown the local or a remote server</b>	Select this option if you want to shutdown (as opposed to restart) either the local server or a named remote server.

Depending on the response option selected, other fields on this window may be enabled. In some cases, the Scripting Wizard will be invoked, allowing you to generate the appropriate response automatically.

 No scripting knowledge is required to use the Scripting Wizard, although the generated script can be further enhanced or edited if required.

For more information see the “Sentry-go Scripting Wizard” guide.

## Command

Depending on the option selected above, you should enter the name of the file, script, or command that you wish to run in response to the triggered error in the next field.

 The files entered must be available to & accessible by the local server. Always specify the full path, including drive or UNC path as appropriate.

If the file is located on another server (not recommended in case it becomes unavailable), always specify the full UNC path with the file. Do not use mapped drives.

 Commands, paths & filenames can contain environment variables (in the format %<variable>%) or Sentry-go place markers (e.g. <\$\$ERROR>) if required. These will be automatically translated at runtime, prior to running the response. For more information see the “Sentry-go Place-markers” guide.

Example scripts can be found on-line at <http://www.sentry-go.com/sentry-go-scripting-library.aspx>.

For checks that automatically invoke the Scripting Wizard, the Wizard itself will generate a filename and pre-populate it once complete.

## Scripting Wizard

To manually invoke the Sentry-go Scripting Wizard, click the “Scripting Wizard” button. This can be used to generate a new script based on an existing template file (without the need for programming or scripting knowledge) or access information, where available, for an existing script.

For more information, see the “Sentry-go Scripting Wizard” guide.

Example scripts can be found on-line at <http://www.sentry-go.com/sentry-go-scripting-library.aspx>.

## Download

Click the “Download” button to launch your web browser & connect to the on-line Sentry-go scripting library. Example scripts & templates can be accessed & downloaded from here.

## Parameters

Enter any parameters or command-line switches you wish to pass to the file or script as part of the response. If no parameters are needed, leave this field blank.

 Parameters can contain environment variables (in the format %<variable>%) or Sentry-go place markers (e.g. <\$\$ERROR>) if required. These will be translated at runtime, prior to running the response.

For more information, see the “Sentry-go Place-markers” guide.

## Connect using dialup before running response ?

If the monitor should connect to dial-up networking before running the response, tick the next option. For more information, see the “Sentry-go Dial-up Networking” guide.

## Disable timeout checking ?

If the response may take time to run and you don't want it to timeout, tick the next option to disable it.

 By default, Sentry-go will monitor the action taken. If it takes an excessive amount of time to complete, the system automatically terminates the job and fails the check.

## Do not perform check again after response is run ?

If you don't want Sentry-go to re-run the check after the response has been run, tick the next option (where applicable).

 Often, you will want to verify the check to see if the response has corrected the fault that triggered the alert originally. To do this, leave this option unchecked. However, there may be some cases when you do not wish to do this – for example, when a check is used to trigger a related job that will not affect the monitored item. To do this, tick this option.

This option is useful for options such as file conversions. The file monitor can trigger an alert when a new item is added to a directory, and the resulting response used to perform a file conversion – e.g. Word to PDF. In this case you don't wish to run the check again and so you would tick this option.

## Run command/script as a different user ?

If you wish to run the response as a named user, click the “User” button and enter user details in the resulting window.

-  By default, any response will be run as the user running the Sentry-go monitoring service. Typically this will be the local system account which has administrative privileges to the local machine only. If you wish to run the response under a specific account – e.g. a domain account with specific permissions or an account which has access to network resources etc., you can specify a named user here.

For more information, see the “Running File or Script as Named User” guide.

## Auto-response considerations

When configuring auto responses, the first & most obvious thing to consider is what action, if any, the monitor should take. Sometimes, this will be obvious – a command or specific action. On other occasions, maybe you simply want to be informed of the error & don’t want anything else to happen.

For example, in these situations ...

- When the machine runs low on memory
- When the runs more than the expected number of processes
- A higher than expected number of logon attempts fail

... there’s not really anything specific to do except investigate why they occurred and their longer term resolution. If however, you can & do choose to configure an automatic response, you can select a number of different types.

### Running a pre-configured response

Sentry-go monitoring components come pre-configured with a number of actions, relating to the task being monitored. They’re designed to make life easy and often you simply need to select the appropriate action in the “Response” tab and let the monitor do the rest.

Examples include ...

- Restarting or continuing a stopped or paused service.
- Terminate a blocking SQL query.
- Pausing a queued document that is considered too large for the printer.

### Running an “Implicit” Script

“Implicit” scripts are selectable responses that actually cause a script to be generated to perform the required functions. The Sentry-go Scripting Wizard is automatically invoked in this case, allowing the file to be generated without the need for coding or script knowledge.

For more information see the “Sentry-go Scripting Wizard” guide.

## **Running a Windows command**

Alternatively, there may be a command you would use on the command line as a first resolution. These are direct commands you might run on the command line having logged on to the server. For example ...

- Net start <ServiceName>.
- Net use <Mapped Drive Information>

## **Write & run a batch file containing one or more commands**

Batch files allow a group one or more Windows commands or applications to be grouped & run in sequence. Basic logic can also be performed within the batch file.

## **Write & run a script (e.g. VB Script) containing logic to perform the required response**

This is often considered one step up from the batch file and allows far greater programmatic control and access. If you're a developer, or know scripting such as VBScript, then you can perform a vast array of functions and access even more through extendible objects such as Microsoft's CDO (for e-mail), the File System Object (FSO for files) and ADO for database access.

Complex, parameterised & conditional logic can be performed using scripting.

## **Reboot the local (monitored) server**

Often a last resort, there may be occasions when the primary role of the server has been compromised to such an extent that the first thing you'd do is restart Windows. A system reboot asks Sentry-go to restart the local server – the server on which the monitor is running.

To guard against a never ending loop, in-built logic within Sentry-go helps to ensure that the same check failure doesn't cause a system reboot to be repeated.

## Consider All the Options

Although the automatic response options given above are designed to be easy to use and quick to configure, it's worth considering all the options. Sometimes, for example, you may want to try one thing and if that fails, do something else. For these scenarios, the script is an obvious candidate, even if one of the actions can easily be performed using a different technique.

For example, if a specific service fails, you might want to restart other related services at the same time, checking their status as you go. A simple script could easily be written to stop, then start each in turn (or stop all, then restart them in sequence).

## Some Dos & Don'ts

### **Make sure all paths are local, or at least accessible from the monitoring server.**

It's the Sentry-go monitoring service that runs the response & so all files must be accessible to it. This includes any files or commands contained inside a batch file or script that's being called.

Although you may be configuring the system remotely, remember that paths must be local - or at the very least accessible to the local server over the network. The use of local paths is recommended.

### **Avoid mapped drives for network paths – use UNC names instead**

Always configure paths using a full UNC path - [\\Server\Share\Folder](#) etc. and avoid referring to a network drive using a mapped name.

For example, you may map the M: drive to a particular server but when referring to a file or folder on that server, use the full server & share name (in UNC format) and avoid using M: in the path.

Sentry-go performs its monitoring & auto-responses from a Windows service. Although this typically as local administrative permissions, it will often have no access to the mapped drives your user session has. Most mappings are created by Windows Explorer when you logon; Explorer doesn't run for a service and hence no mapping will be available.

### **Use fully qualified paths**

Within the file or script, ensure any files, paths or required EXEs are qualified with the full path. This ensures that wherever the script is run from (its current directory), the path will still be valid.

## Consider the “current directory”

Some utilities expect to run from the current directory, when run from batch files. To ensure the correct path is set, "CD" into the appropriate directory before running the utility. For example ...

```
CD \new-path\new-folder  
MyCommand.exe
```

Also, note that if the target directory is (or could be) different from the drive on which the monitor is running you should change to the new drive before CD'ing into the target directory. In other words, don't attempt to change drive within the CD command. If you do, the command may not work as you expect. For example, to ensure you're running on the D: drive ...

```
D:  
CD \new-path\new-folder  
MyCommand.exe
```

## Remember who's running the script (it's probably not you)

When you run the script from a command prompt or Start/Run, you are implicitly running it under your own Windows user account.

When you run it as a response, however, by default it's the user running the monitoring service that's in charge. This can have an impact on the way it works (or doesn't work).

Specifically, your user ID is most likely to be a domain user with access to both local & network resources. By default, however, the monitoring user is the Local System user – which has Administrative privileges locally but few or no access to the domain or network. This may be important, depending on what the script is going to do.

If your script is likely to need domain access ...

- Run the response as a named (domain) user.
- Run the monitoring service as a domain user (with local administrative privileges).

See the “Running File or Script as Named User” guide for more information.

## Access network resources carefully

If your response does rely on a network resource, configure your response carefully. There are a couple of reasons for taking care ...

- The network or resource may not be available when you attempt to run the response. If this happens, the response will fail, even though the actual solution would, if run, have resolved the problem.
- The user running the response must have access & permission to access the network resource.

See the “Running File or Script as Named User” guide for more information.

## **Configuring the User ID to run the script**

If you can't, or don't wish to use the Local System account, it is recommended that a dedicated domain user account is configured for use by Sentry-go. This way, you can ...

- Configure it to run Windows services (required if running the Sentry-go monitoring service).
- Grant it the appropriate permissions required on the domain.
- Grant it Administrative privileges on the local server

## **Remember where the response is being run from**

The response is always run by the monitoring service – i.e. on the machine running the Sentry-go monitor that detected the fault. Even if remote UNC are used, it's the local server that's running the file.

It is strongly recommended that responses are run locally. However, if remote access/running is required, the script itself must be coded to access remote resources.

See below for more information.

## **Consider what your file or script will access**

In some cases, your script may rely on other technologies or installed applications to run (MAPI for example). This is normally fine, but if these technologies also rely on others, or they require a logged on user (or at least an available user profile) you will need to ensure the following ...

- Check that each component is installed on your server.
- Check that the user running the response has access to, and can run these components.

Also, some technologies assume a user is running them from a desktop window. With Sentry-go, there's no desktop window as the service itself is running the job and by default, no user profile will be available (as the local system user doesn't have an interactive session).

To overcome this, either configure the response to run as a domain user (this will load the user's domain profile), or run the Sentry-go monitoring service as a domain user – with the appropriate permissions locally & on the domain (which will also cause their profile to be loaded when the monitor is run).

## **Don't display an interface (GUI)**

Windows services (such as the monitoring service) typically run without a user interface or Windows session. In fact, most likely of all, no user will be logged on directly to the server.

With this in mind, you must avoid calling or running applications which contain a user interface (including message boxes etc.) as this window ...

- Will not be seen (even if you are logged on to the server)
- Cannot be accessed (or closed).

## Ensure the script doesn't take too long to run

After a response is initiated, Sentry-go will wait for it to complete before proceeding. It determines this by monitoring the job itself and, once it closes (and therefore exits), re-checks the monitored item.

During this time, the monitoring component calling the response will wait (as other checks may be affected by the result) and hence the quicker the response, the faster the monitor will be at performing other checks.

To protect itself against a hung response, Sentry-go, by default will only wait so long for the script or file to run. If it hasn't completed within this period, it will time out and the response deemed to have failed.

- Try to ensure the response works as quickly & efficiently as possible.
- If a longer response must be run, you can increase the default timeout, so that the monitor allows additional time (though this is not recommended as it could delay other monitoring tasks).

## Local vs Remote Responses

Sentry-go is designed to run on the server being monitored. There are a number of reasons for this, for example ...

- Local monitoring is much more efficient than performing it across the network
- There is no network overhead
- There is no reliance on the network. If a network link should fail, it will not impact on the monitor

For the same reasons, we strongly recommend that response files & scripts are located and run directly on the local server – i.e. the server where the monitoring service is running. This ensures that resources should be available & the monitor will have access to them.

### Running Remote Responses

Although the use of local responses is recommended, there may be times when it's more appropriate to perform a remote action. For example ...

- The event that generated the error was caused by an issue on a remote server.
- The monitor check itself was checking something remotely across the network – e.g. via a script.

In these cases, where corrective action may be needed elsewhere, you might consider using a remote response. How you achieve this will depend on the facilities available to you, the version of Windows being run and the access you have available from the monitoring machine.

Below are some examples of how this might be achieved. *Note that these are just examples designed to give you an idea of what's involved. They are designed to be used without further amendment & development.*

## Use Windows Management Instrumentation (WMI) Objects

Windows Management Instrumentation or WMI is an optional technology available to Windows. It provides a number of objects (classes) that allow you to interact with different parts of the software & hardware system. More importantly, it also allows you to do this on the local or remote machine.

For example, you could stop/start one or more Windows services - or even restart the Windows installation remotely, you can use a script containing WMI commands. WMI must be installed on the monitoring & target servers.

```
` Declarations
Dim strWQL
Dim objWMI
Dim colResult
Dim objWinService
Dim intResult
Dim strTargetServer
Dim strTargetService

` Error handling
On Error Resume Next

` Set target server name & service
strTargetSvr = "Your Server name"
strTargetSvc = "Your Service name"

` Access WMI
Set objWMI = GetObject("winmgmts:\\\" & strTargetSvr & "\root\cimv2")
If Err <> 0 Then

    ` Failed
Else

    ` Query WMI
    strWQL = "SELECT * FROM Win32_Service WHERE Name = '\" & strTargetSvc & '\""
    Set colResult = objWMI.ExecQuery(strWQL)

    ` Check results
    If colResult.Count > 0 Then

        For Each objSvc in colResult

            ` If we want to start it
            If objSvc.Started = False Then

                intResult = objSvc.StartService
                If intResult <> 0 Then

                    ` Failed
                    End If
                End If

            ` If we want to stop it
            If objSvc.Started = True Then

                intResult = objSvc.StopService
                If intResult <> 0 Then

                    ` Failed
                    End If
                End If
            End If
        Next
    End If

    ` Cleanup
    Set objWMI = Nothing
End If
```

## Use the WMI component to remotely execute a command file or script

Using the above WMI components, you can also connect to a remote machine and run one or more commands, files or scripts. Using this feature you could host batch files & scripts on the target server and call them from the monitoring machine as required.

To do this you would need to ...

- Connect to WMI on the remote server
- Access the Win32\_Process object of WMI
- Ensure the user running the script has permissions to access & run remote commands through WMI on the remote server
- Or specify user credentials within the script to connect to WMI with a given user ID before starting the process.

```
` Declarations
Dim objWBEMLocator
Dim objWMI
Dim objProcess
Dim objStartup
Dim objConfig
Dim strTargetSvr
Dim strTargetUser
Dim strTargetPassword
Dim strTargetCommand
Dim intResult

` Constants
Const cWBEM_IMPERSONATE = 3
Const cWBEM_PKT_PRIVACY = 6
Const cREMOTE_EXEC = -1

` Error handling
On Error Resume Next

` Set target details
strTargetSvr = "Your Server name"
strTargetSvc = "Your Service name"
strTargetUser = "Your user name"
strTargetPassword = "Your password name"
strTargetCommand = "The command to run (must be accessible to remote server)"

` Access WBEM
Set objWBEMLocator = CreateObject("WbemScripting.SWbemLocator")
If Err.Number <> 0 Then

    ' Failed

End If

` Connect to QMI
Set objWMI = objWBEMLocator.ConnectServer _
    (strTargetSvr,"root\cimv2", strTargetUser, strTargetPassword)
If Err.Number <> 0 Then

    ' Failed

End If

objWMI.Security_.ImpersonationLevel = cWBEM_IMPERSONATE
objWMI.Security_.AuthenticationLevel = cWBEM_PKT_PRIVACY
```

```

' Configure the new process
Set objStartup = objWMI.Get("Win32_ProcessStartup")
If Err.Number <> 0 Then

    ' Failed

End If

Set objConfig = objStartup.SpawnInstance_
If Err.Number <> 0 Then

    ' Failed

End If

objConfig.ShowWindow = SW_NORMAL

Set objProcess = objWMI.Get("Win32_Process")
If Err.Number <> 0 Then

    ' Failed

End If

intResult = Process.Create (strTargetCommand, NULL, NULL, intProcessID)
If Err.Number <> 0 Then

    ' Failed - result code indicates reason

End If

' Cleanup
Set objWMI = nothing
Set objWBEMLocator = nothing

```

### **Use 3<sup>rd</sup> party remote execution technology**

In addition to WMI, other remote execution technologies are available with Windows and through 3<sup>rd</sup> parties. For example, "RSH" in Windows XP. These can often be used from the command line, and therefore batch files & scripts to remotely execute logic remotely.

## More Information

If you need more help or information on this topic ...

- Read all [papers/documents on-line](#).
- Watch [demonstrations & walkthrough videos on-line](#).
- Visit <http://www.Sentry-go.com>.
- Contact our [Support Team](#).



*Sentry-go, © 3Ds (UK) Limited, 2000-2013  
East Molesey, Surrey, United Kingdom  
T. 0208 144 4141  
W. <http://www.Sentry-go.com>*