

Custom Alert Response Scripts

RTO PinPoint can execute scripts in response to an alert event. Any script supported by Windows Script Host (including VBScript and Jscript) can be specified to perform additional custom tasks when RTO PinPoint generates an alert.

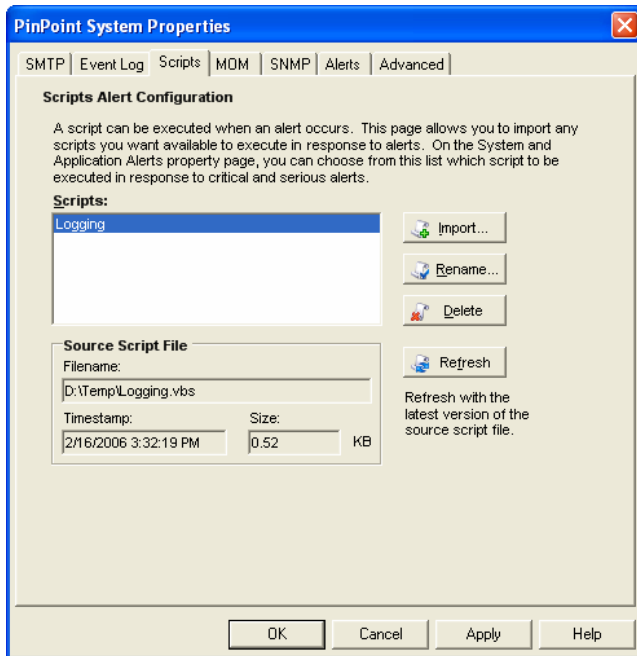
Scripts can be configured globally in the Alerts tab on the RTO PinPoint System Properties page or for a specific application in the Alerts tab on the Application Properties page. Application specific scripts override the global script settings and apply to all metrics for that application and its servers.

The server on which the script is executed depends on the scope of the alert. Any alert associated with an application or application metric will be run on the RTO PinPoint Analysis Engine server. Any alert associated with a server or server metric will be run by the RTO PinPoint Agent on the server that caused the alert.

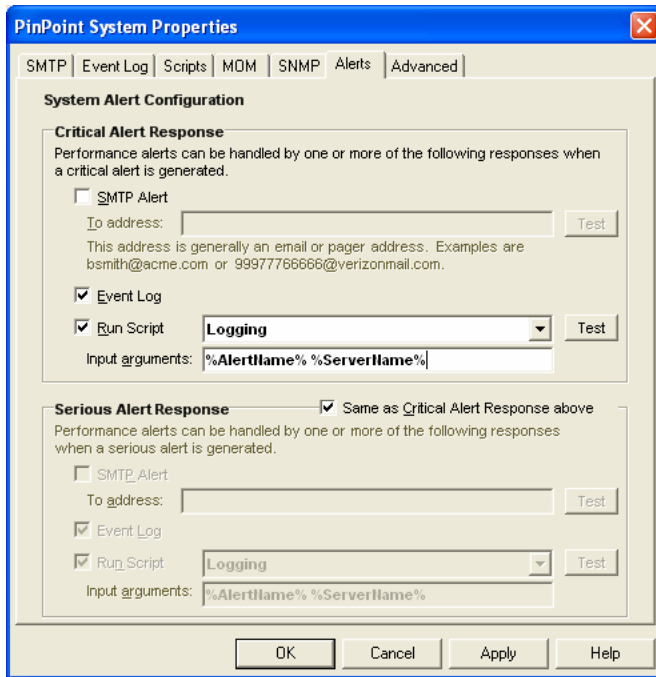
Adding Scripts

To add a script to RTO PinPoint you must first import the script and then select that script as a response for a critical or serious alert.

To import the script go to the RTO PinPoint System Properties page (below) by right clicking the RTO RTO PinPoint root node in the tree on the left on the Console. Move to the Scripts tab and click the Import... button to select your script file. Imported script files are copied into the RTO PinPoint system so when scripts are updated you must return to this property page and click the Refresh button to re-import the latest changes.



To select an imported script as the response to an alert go to the Alerts tab in the RTO PinPoint System Properties page (below). Check the Run Script checkbox under either Critical Alert Response or Serious Alert Response and select the appropriate script from the adjacent combo box. You can also specify input arguments to the script using constant values or RTO PinPoint script variables described in the next section. Clicking the Test button will cause the script to be executed immediately on the RTO PinPoint Analysis Engine server using test data in place of any script variables.



The image shows a screenshot of the 'PinPoint System Properties' dialog box. The 'Advanced' tab is selected. Under 'System Alert Configuration', there are two sections: 'Critical Alert Response' and 'Serious Alert Response'. In the 'Critical Alert Response' section, 'SMTP Alert' is unchecked, 'Event Log' and 'Run Script' are checked. The 'Run Script' dropdown is set to 'Logging' and the input arguments are '%AlertName% %ServerName%'. The 'Serious Alert Response' section has a checkbox 'Same as Critical Alert Response above' which is checked. It also has 'Event Log' and 'Run Script' checked, with 'Run Script' set to 'Logging' and input arguments '%AlertName% %ServerName%'. Buttons for 'OK', 'Cancel', 'Apply', and 'Help' are at the bottom.

Any script settings configured on the RTO PinPoint System Properties page above will be applied globally to all applications and servers unless overridden at the application level via the Application Properties page.

Script Input Arguments

The following variables can be used as input arguments whenever a script is executed. You can use these variables to implement different logic paths in the script depending on which alert happened, when it happened, or where it happened.

Script Variable	Description
%AlertTime%	The time that the alert was raised
%AlertName%	The name of the alert
%AlertDescription%	The description of the alert
%AlertInstance%	The name of the specific instance that generated the alert (only valid for certain metric alerts)
%ReportURI%	The UNC path to the root cause analysis report (only valid for alerts on the application performance score)
%ApplicationName%	The name of the application that generated the alert
%ServerName%	The name of the server that generated the alert (only valid for server metric alerts)

Sample Script

The following simple script writes a log entry to the file c:\alert.log each time it is run. The log entry consists of the text "a log entry" on the first line followed by the value of the first six input arguments on subsequent lines.

```
Dim FileSystemObject, TextStream

Set FileSystemObject = CreateObject("Scripting.FileSystemObject")
Set TextStream = FileSystemObject.OpenTextFile("C:\alert.log", 8, True)

TextStream.WriteLine "a log entry"
TextStream.WriteLine wscript.arguments.item(0)
TextStream.WriteLine wscript.arguments.item(1)
TextStream.WriteLine wscript.arguments.item(2)
TextStream.WriteLine wscript.arguments.item(3)
TextStream.WriteLine wscript.arguments.item(4)
TextStream.WriteLine wscript.arguments.item(5)

TextStream.Close
```

Running this script in response to an application performance index alert using the following input arguments produces the log entry below.

Input arguments: %AlertTime% %AlertName% %AlertDescription%

```
a log entry
5/15/2007 3:15:20 PM
The application performance index is low
The Application Performance Score is below an acceptable level
```