Running PowerShell Scripts with Uptime Infrastructure Monitor

Introduction

Powershell is a interface for interacting with Windows systems in a scripted or automated method similar to what can be done with Bash within the Linux world.

One of powershell's biggest features/paradigms is that it treats everything as an Object instead of as Files or Text like traditional scripting languages(ie Bash, Perl, Python etc)

This object orientated approach means that powershell also handles it's output differently then the traditional standard out/standard in conventions. From the Uptime Infrastructure Monitor perspective this means we need to use a slightly different approach when triggering powershell scripts as Custom Monitors or Action Profiles. As well when using powershell scripts as custom commands via the Windows Agent.

Another thing to keep in mind with Powershell in general is that it handles 'white space' and in paths differently then the normal Windows command prompt. This is also complicated by powershell's behaviour to just echo out any strings wrapped in double quotes. Which is the traditional way for handling white space within Windows Paths. The easiest way to avoid this by placing your scripts in a path that doesn't have any white-spaces, like in the examples below. But if you do need call a script from a path that contains white-space you'll need to use the escape back-tick (ie. `) before every white space character.

One last thing to keep in mind with Powershell scripts when compared to traditional scripting languages, is that also has the idea of an 'Execution Policy', that controls what scripts or configuration files can be used. By default Powershell runs under the 'Restricted' execution Policy. For more details about Powershell's execution policys in general please refer to 'about_Execution_Policies' from Microsoft or the 'Set-ExecutionPolicy cmdlet' which allows you change these behaviours.

Monitoring Station Side:

Obviously in order to leverage powershell scripts on your Uptime Infrastructure Monitor monitoring station, it needs to be running on a Windows system.

The first step to setting up your Powershell script for use with Uptime Infrastructure Monitor is to create a simple wrapper script in batch similar to the 'mypowershell.bat' example below. The '<NUL ' at the end of the line tells the Uptime Infrastructure Monitor JVM to stop waiting for more output from the ps1 script and return the contents to Uptime Infrastructure Monitor.

mypowershell.bat

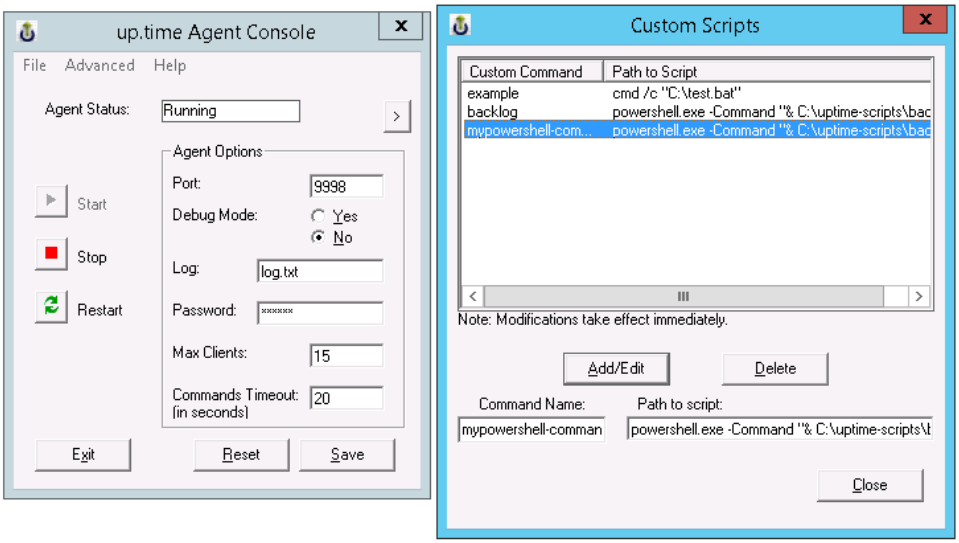
```
@ECHO OFF
powershell.exe -noprofile -Command "& 'c:\uptime-scripts\backlog-monitor.ps1' " <NUL
```

Once you have your batch wrapper script created, you can then use the path to the .bat file in the various 'Script Name' fields within Uptime Infrastructure Monitor's custom monitors or as part of Alert/Action Profiles just like you would any other script.

Agent Side:

Setting up powershell script as custom commands for the Windows Agent is easier then the monitoring station example above, as we don't need to create the .bat wrapper script, and instead can provide the powershell.exe flags/arguments directly in the Agent Console's 'Path to Script' field.

Here's an example of a powershell script setup a Custom Script within the Windows Agent:



Here's the full contents of the 'Path to Script' field in the screenshot above.

```
powershell.exe -Command "& C:\path\to\your.ps1"
```

Keep in mind that in addition to setting your 'Custom Command'/'Path to script' values, you'll also need to set an Agent Password in the agent console as well. Once you have your Powershell script setup in the Agent Console, you'll likely want to use either the 'Custom Remote Monitor' or 'Custom Remote Integer' plugin monitors to setup a service monitor to run your script.