Automation: Service restart script for Windows Application

|  |  |
| --- | --- |
| **Version :** | 1.0 |
| **Date :** | 08/11/2016 |
| **Status :** | Draft |
| **Classification :** | Internal |

Contents

[1. Overview: 3](#_Toc466035152)

[2. What the script do? 3](#_Toc466035153)

[3. Pre\_Requisites: 3](#_Toc466035154)

[4. How the script works? 4](#_Toc466035155)

[5. Error and Exception handling: 4](#_Toc466035156)

[6. Scheduling part: 5](#_Toc466035157)

[7. Post verifications/Challenges: 5](#_Toc466035158)

[8. References 6](#_Toc466035159)

[8.1 Document Control 6](#_Toc466035160)

[8.2 Version Control 6](#_Toc466035161)

# 

# Overview:

Automation solutions enable IT professional to leverage a centralized automation engine to interact with any target System; they are flexible and extensible services that can be leveraged for almost any task that can be imagined.

Windows service general script was written to do multiple services in different target servers depending on the requirement.

# 2. What the script do?

Script takes action to be performed, server name and services list text file as parameter and performs the action on all services listed in text file in target server. We can perform multiple actions on services like start, stop or restart.

When script is triggered script will do below action

1. Check the current status of the service and write it to log
2. Start/Stop/Restart service based on user passed parameter
3. Check the status of the service after auctioning and write it to log file

**Usage:**

* powershell.exe -file <Script Path> <Action> <Country Code> <Application Text File>

Where

Script Path: Centralised path of the script

Action: start/stop/restart

Country Code: 2 letter country code to restart the service for particular country (eg. AU, SG, CN etc)

Text File: Input file containing Country code respective server and services list in order which needs to be restarted

# 3. Prerequisites:

1. Generic service account with administrator access should be created where services require to be restarted.
2. Generic service account should be registered.

Presently Control-M agent is registered prior to run job to schedule job from Control-M;

Further might work with System-centre.

# 4. How the script works?

Script uses the PowerShell cmdlet related to handling services in windows servers such as get-services, stop-service, Start-service and restart-service etc.

Once script triggered script does the initial validation on parameter passed such that whether server is reachable, does input file exist etc.

After basic initial validation, script start, stop or restart services based on the action passed with PowerShell cmdlet.

Script is generic for all windows servers running in Global domain and can be reused by multiple applications.

Scripts take actions dynamically depending on the parameter we passed.

We need to keep the services names stored in text file (Can be converted to XML in future) for each applications in format of <CountryCode><ServerName><ServiceName>, Script take server name and services list text file as input parameter.

# 5. Error and Exception handling:

Script must be tuned to handle common error as well as unexpected end of operation.

Basic error handling contains below scenarios;

|  |  |
| --- | --- |
| **RETURN CODE** | **ERROR DESCRIPTION** |
| **8** | ERROR: Unable to get service details for $ServiceName in server $ServerName |
| **4** | Service $ServiceName in server $ServerName could NOT be stopped.Status of service is STARTED.Please stop service manually. |
| **2** | Invalid action parameter passed.NO ACTION TAKEN BY SCRIPT. Valid parameter for actions are start/stop/restart. Parameter Passed: $Action |
| **0** | Job completed successfully with return code: $ReturnCode |

# 6. Scheduling part:

Currently, Scheduling is done in Control-M, Further planning to be implemented in Microsoft- System Centre.

Parameters that might be considered during scheduling include:

* Job priority.
* Compute resource availability.
* License key if job is using licensed software.
* Automatic restart and recovery in event of failures.
* Alerting and notification to operations personnel.
* Generation of incident reports.
* Execution time allocated to user.
* Number of simultaneous jobs allowed for a user.
* Estimated and Elapsed execution time.
* Availability of peripheral devices.
* Job and File dependency.

# 7. Post verifications/Challenges:

1.       Have the logic within the script to check the status of the service in remote server after start/stop/restart to make sure job done its intended work.

2.       Alerting to support team in case of any error/warning.

3. Post restarts application level verification.

# 8. References:

The following documents have been considered in are referenced in this document:

| ID | Document Title | Version |
| --- | --- | --- |
| 001 | Windows service restart script Test Scenario | 1.0 |
| 002 | Input file | 1.0 |

## 8.1 Document Control

 

## 8.2 Version Control

| Version | Date | Change Description | Author |
| --- | --- | --- | --- |
| 1.0 | 08/11/2016 | Initial Draft | Gowthami M R |
|  |  |  |  |
|  |  |  |  |