Server Standup Script

Contents

[Overview 2](#_Toc418514189)

[Governance 2](#_Toc418514190)

[Design 2](#_Toc418514191)

[Features 2](#_Toc418514192)

[Running the Script 3](#_Toc418514193)

[XML Configuration file 4](#_Toc418514194)

[Logging 4](#_Toc418514195)

[Folders 4](#_Toc418514196)

[EnvVars 4](#_Toc418514197)

[LocalRights 5](#_Toc418514198)

[Bindependency 5](#_Toc418514199)

[ServerManager 5](#_Toc418514200)

[IIS 6](#_Toc418514201)

[MSDTC 11](#_Toc418514202)

[ServiceConfig 12](#_Toc418514203)

[Reg 13](#_Toc418514204)

[MSMQQueue 13](#_Toc418514205)

[Volume 15](#_Toc418514206)

[EVLogs 16](#_Toc418514207)

[Removefile 17](#_Toc418514208)

[Filesystem 17](#_Toc418514209)

[AppConfig 19](#_Toc418514210)

[WindowsFirewall 19](#_Toc418514211)

[MSSQL 20](#_Toc418514212)

[SpecialConf 20](#_Toc418514213)

[SearchAndReplace 21](#_Toc418514214)

[TextInsert 21](#_Toc418514215)

[FailoverClustering 21](#_Toc418514216)

[Desired State Configuration (DSC) 22](#_Toc418514217)

[Notes 22](#_Toc418514218)

## Overview

The Server Standup Script (STIG-ServerStandup.ps1) is a data driven (XML) library that is intended to perform all aspects of Windows server configuration. The intent is to ensure that any/all server configuration is handled via XML files (i.e. Software Defined Systems) that are committed to source control (Team Foundation Server / TFS) and therefore follow best practice for SCM.

### Governance

Data driven / XML based

All code committed to Version Control (TFS)

Modular

Libraries

Promote changes same as SDLC

Test changes prior to commit

Import validated changes into uDeploy for DML distribution

### Design

Main script: STIG-ServerStandup.ps1

Library: ./Includes/

Powershell Scripts specific to the STIG-ServerStandup.ps1 script must be placed within the *Includes* folder, and prefaced with “STIG-“. Additional library functions not directly related to STIG standup script (however still used by the script) must be also be placed in the *Includes* folder, and prefaced with “LIB-“. “LIB” scripts are generic in nature, and are intended to be used by ANY script (“. source” separation of certain subsets of the scripts by the “LIB-\*.ps1” name)

## Features

As of version 1.50, The Server Standup Script (STIG-ServerStandup.ps1) is an extremely versatile script that can perform the following actions:

* Download files and folders locally
* Format and partition local and iSCSI disks
* Re-address/ Relocate drives
* Install .NET 4.0 / 4.5, Java JRE, Octopus Tentacle, Webdeploy
* Install Windows roles and services
* Configure IIS App Pools, SSL Certs Websites, Web Apps, Virtual Directories, Enabled protocols, Enabled authentication, Custom response headers, IIS Compression and IIS Output caching.
* Configure Windows Failover clusters
* Install and configure Apache Tomcat and Apache Solr
* Install MS SQL
* Configure MSDTC and MSMQ
* Configure Windows services
* Set file system permissions (NTFS)
* Create and update Windows Event Logs
* Set Registry keys
* Integration with Desired State Configuration (Windows 2012/Powershell 4 and later)

All of this is done through an extensive library of scripts located in the Includes folder, which the main STIG-ServerStandup script calls during execution.

The script can be used to put a host at an end-state ready for deployment based on a completely customizable XML file.

## Running the Script

Because you need to copy the libraries and supporting XML and INI files to the local server/path it can be a bit inconvenient running the STIG-ServerStandup.ps1. However there are two convenient ways to run the standup script. Some of the XML and INI files have tokenized strings in them that need to be replaced at runtime with desired values.

**uDeploy**

See the uDeploy documentation at <http://insite.bridgepoint.local/collaborate/scm/SCM%20WIKI%20TESTING/Udeploy.aspx> "Using uDeploy for specific tasks" for how to run from uDeploy.

**Powershell**

You can run the helper script Get-ServerStandupFiles.ps1 which can be found in the DML. This script will copy the relevant files from the DML to a compliant path structure on the local server, if a token replacement definition file is provided it will then perform the token replacement, and then will run the startup script with the desired XML file (unless you wish to simply download the files, for example if you want to run the script in the ISE or maybe swap out some library files before running; then you can specify not to execute automatically).

Here are some examples of running the script without using uDeploy:

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -XMLFile CampusVuePortal\CampusVueAPPNexus.xml -Test

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -XMLFile CampusVuePortal\CampusVueConnectorNexus.xml

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -XMLFile CampusVuePortal\CampusVueTerminalNexus.xml

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -TokenFile c:\temp\sqldevtokenfile2.dat -XMLFile SQL\CampusVueSQLNexus.xml

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -XMLFile WebServices\waypointtableau.xml

\\10.13.0.206\scratch\DML\Scripts\Get-ServerStandupFiles.ps1 -TokenFile c:\temp\sqldevtokenfile2.dat -XMLFile SQL\Win2008.BASE2008.MSSQL.xml -test

You can cut and paste one of these lines directly into the Powershell window on the target computer and the download and execution will begin. Note that because of our GPO running the script from a network source will require a confirmation from you.

## XML Configuration file

By Default, the STIG-ServerStandup.ps1 file needs an XML file to direct its actions. Included with the script is a params.xml file that should ONLY be used as reference. The Root XML Element is <params>. Every element referenced here is a sub-element of that.

### Logging

<Logging State="Enabled" Name="STIG-Standup[TimeStamp].log" />

* State: Enable or Disable Logging
* Name: The name of the logfile to create. [TimeStamp] is a variable used by the script

### Folders

<Folders>

<!--Leave the folder alone. Also, don’t remove it from any version of this file-->

<Includes>Includes</Includes>

<Temp>tmp</Temp>

</Folders>

* As the comment suggests, leave this XML sub element alone, but ensure that it’s in every version of the xml file you deploy.
* This controls the module includes folder, and the temp folder to create.

### EnvVars

<EnvVars>

<Var Action="Add|Set|Del" Variable=":string:" Value=":string:" Target="Machine|User" />

</EnvVars>

* When action is "Add" then the specified variable/value will be appended to the environment variable unless that exact value already exists in the variable, in which case nothing will change.
* When action is "Set" then the specified variable/value will replace an existing variable.
* Both actions will create the variable if it doesn't already exist.
* When the action is delete then the specified variable will be deleted.

### LocalRights

<LocalRights Action="Add" User="bridgepoint\sql\_server" Right="SeLockMemoryPrivilege" />

### Bindependency

<Bindependency>

<ServerBinRoot Name="\\10.13.0.206\scratch" />

<Folder Action="Execute" ExeName="SQLInstall.exe" PathFromBinRoot="DML\Microsoft\MSSQL\10.0\Folder">

<Argument>/q</Argument>

<Argument>/norestart</Argument>

</Folder>

<File Name="dotNetFx40\_Full\_x86\_x64.exe" Action="Execute" PathFromBinRoot="DML\Microsoft\DotNet\_Framework\v4.0\dotNetFx40\_Full\_x86\_x64.exe" ProductIsInstalled="dir &quot;C:\Windows\Microsoft.NET\Framework64&quot; | ? {$\_.Name -like &quot;v4.0\*&quot;}" RetryMax="2">

<Argument>/q</Argument>

<Argument>/norestart</Argument>

</File>

</Bindependency>

* ServerBinRoot Name: This is the top level location where all files and folders that need to get downloaded are downloaded from.
* Folder/File Action: Add
* Folder/File ExeName: The executable to run after the folder has been downloaded
* Folder/File PathFromBinRoot: The path to the folder using ServerBinRoot as the relative root path.
* Folder/File ProductIsInstalled: A command to run to check if the installation was successful
* Folder/File RetryMax: The maximum amount of times to retry the installer if previous attempt(s) failed
* Argument: any cmdline arguments that need to be ran with the executable.

### ServerManager

<ServerManager>

<includeallSubFeature>

<feature>web-WebServer</feature>

</includeallSubFeature>

<Single>

<feature>web-WebServer</feature>

</Single>

* Includeallsubfeature/Single Feature: Name of the windows feature / role to install. If the feature is under the Includeallsubfeature sub-node, then it will also install all sub items of that feature / role. Single will ONLY install that named feature.

### IIS

<WWWROOT>C:\WWWROOT</WWWROOT>

* IIS / WWWRoot: The top level folder where the IIS Tree will be located. All folders will be created as a subset of this.

#### IIS / RegDOTNETIIS

<RegDOTNETIIS>True</RegDOTNETIIS>

* RegDOTNETIIS: True | False. If set to true, register .NET in IIS.

#### IIS / CreateFolderStr

<CreateFolderStr>

<DIR>Folder1\SubFolder1</DIR>

<DIR>Folder2\SubFolder1</DIR>

</CreateFolderStr>

* IIS / CreateFolderStr / Dir: Create a folder structure under the WWWRoot

#### IIS / SSL

<SSL>

<Cert Action="Copy" Name="ExistingCert.pfx" CertPW="Prompt | ClearTxtPW" SecurePW="SecureStringPW" />

</SSL>

* IIS / SSL / Cert Action: Add | Copy. If copying, ensure that the ssl cert is copied using BinDependency / File; note that the <Temp></Temp> location can be used to store the certificate for import (no need to specify the full path), however, that directory will be deleted upon script completion.
* IIS / SSL / Cert Name: Name of the SSL Cert to create or import
* IIS / SSL / Cert CertPW: Prompt | <Password>: Password of the SSL Cert. If set to prompt, the script will prompt to input the password.
* IIS / SSL / Cert SecurePW: <Secure String password>. See about Powershell about secure strings for more information on this

#### IIS / ManageAppPool

<ManageAppPool>

<Pool Action="Add" NAME="ProspectNotification" NETVer="v4.0" IDType="SpecificUser">

<Auth User="%\_USER\_%" Password="%\_PASSWORD\_%" />

</Pool>

</ManageAppPool>

* IIS / ManageAppPool / Pool Action: Add | Delete
* IIS / ManageAppPool / Pool Name: Name of the pool to create or delete
* IIS / ManageAppPool / Pool NETVer: .NET Version to use “v2.0” | “v4.0”
  + If .NET 4.5 is installed, still use “v4.0”
* IIS / ManageAppPool / Pool IDType: "ApplicationPoolIdentity" | "LocalService" | "LocalSystem" | "NetworkService" | "SpecificUser": Set the application pool Identity type.
* IIS / ManageAppPool / Pool / Auth User: User to run the App Pool under
* IIS / ManageAppPool / Pool / Auth Password: Password for the App Pool user
* IIS / ManageAppPool / Pool / Enable32Bit: Enable 32 bit application (enable32BitAppOnWin64) “True” | “False”

#### IIS / Websites

<Websites>

<Site Action="Add" SiteName="InternalServices" PhysPath="LeadManagement" AppPool="ASP.NET v4.0" SSL="" SSLIP="" SSLPort="" SSLCERTName="">

<Binding action="Add" type="http" address="\*" port="8080" hostname="" />

<Binding action="Add" type="net.msmq" address="" port="" hostname="" BindingInfo="mq-msmq1d" />

</site>

</Websites>

* IIS / Websites / Site Action: Add | Delete
* IIS / Websites / Site SiteName: Name of the site to create/delete
* IIS / Websites / Site PhysPath: Physical path using “WWWROOT” as the relative parent root.
* IIS / Websites / Site AppPool: App Pool to use for this website
* IIS / Websites / Site SSL: True. If set, enables SSL for this site.
* IIS / Websites / Site SSLIP: SSL IP address to use. Use \* for All.
* IIS / Websites / Site SSLPort: SSL Port to use
* IIS / Websites / Site SSLCertName: SSL Cert name to use for this site.
* IIS / Websites / Site / Binding Action: Add | Delete. Create or Delete this binding
* IIS / Websites / Site / Binding type: the protocol of this binding
* IIS / Websites / Site / Binding address: the address for this binding.
* IIS / Websites / Site / Binding port: port to use for this binding.
* IIS / Websites / Site / Binding hostname: HostHeader for this binding
* IIS / Websites / Site / Binding BindingInfo: BindingInfo for this binding.

#### IIS / WebApp

<WebApp>

<App Action="Delete" Path="IIS:\Sites\WebSite1\WebApp2" DeletePhysical="False" />

<App Action=”Add” Path="IIS:\Sites\InternalServices\CommunicationAccess" AppPool="ProspectNotification" />

<App Action="AddNew" Name="WebApp2" ParentSite="Site\_B" AppPool="Site\_A" PhysPath="C:\WebApp2" />

</WebApp>

* IIS / Webapp / App Action: Add | Delete | AddNew
* IIS / Webapp / App Path: IIS Path to convert into a WebApp or Delete
* IIS / Webapp / App DeletePhysical: True | False. Delete the physical path associated with this WebApp (if action is Delete). Defaults to “FALSE”.
* IIS / Webapp / App AppPool: Associate this WebApp with the named AppPool.
* IIS / Webapp / App Name: Name of the WebApp to add. Only used when action is “AddNew”.
* IIS / Webapp / App ParentSite: Name of the parent website. Only used when action is “AddNew”.
* IIS / Webapp / App PhysPath: Full physical path for the named WebApp. Only used when action is “AddNew”.

#### IIS / EnabledProto

<EnabledProto>

<Item Action="Add" IISPath="IIS:\Sites\SiteB" Proto="http" />

<Item Action="Delete" IISPath="IIS:\Sites\SiteA" Proto="net.pipe" />

</EnabledProto>

* IIS / EnabledProto / Item Action: Add | Delete. Add or Delete the defined Enabled Protocol.
* IIS / EnabledProto / Item IISPath: IIS Path of the object to define enabled protocols on.
* IIS / EnabledProto / Item Proto: Enabled protocol to define.

#### IIS / EnabledAuth

<EnabledAuth>

<Item Action="Enable" IISPath="IIS:\Sites\Best" AuthType=" anonymousAuthentication “ />

</EnabledAuth>

* IIS / EnabledAuth / Action: Enable | Disable
* IIS / EnabledAuth / IISPath: IIS Path of the object to define
* IIS / EnabledAuth / AuthType: digestAuthentication | anonymousAuthentication | iisClientCertificateMappingAuthentication | basicAuthentication | clientCertificateMappingAuthentication | windowsAuthentication. Authentication type to enable to disable.

#### IIS / VirtDir

<VirtDir>

<VD Action="Add" Name="Share" PhysPath="D:\www\SiteA\Share" IISPath="IIS:\Sites\SiteA\Share" />

</VirtDir>

* IIS / VirtDir / VD Action: Add. Add a virtual Directory. Mandatory.
* IIS / VirtDir / VD Name: Name of the virtual directory to create
* IIS / VirtDir / VD PhysPath: Physical of the Virtual Directory to create
* IIS / VirtDir / VD IISPath: IIS Path of the Virtual Directory to create

#### IIS / ClientCache

<ClientCache>

<Item Action="Add" IISPath="IIS:\Sites\My Site\PathA" Control="NoControl" />

<Item Action="Add" IISPath="IIS:\Sites\SiteA\PathB" Control="DisableCache" />

<Item Action="Add" IISPath="IIS:\Sites\SiteC\PathD\InE" Control="UseMaxAge" ExpiresDays="23" />

<Item Action="Add" IISPath="IIS:\Sites\Default Site B" Control="UseExpires" ExpiresDate="" />

</ClientCache>

* IIS / ClientCache / Item Action: Add. Add a Client Cache Directive
* IIS / ClientCache / Item IISPath: IIS Path where to create the Client Cache Directive
* IIS / ClientCache / Item Control: NoControl | DisableCache | UseMaxAge | UseExpires.
  + NoControl: Instruct the client to not cache content
  + DisableCache: Disable any header cache information
  + UseMaxAge: Set the custom header to expire cache in 5 days. Uses ExpiresDays
  + UseExpires: Sets the custom header to expire cache in X Days from (Today). Uses ExpiresDate
* IIS / ClientCache / Item ExpiresDays: Number of days to expire the cache in. Only used when setting the Control property to UseMaxAge
* IIS / ClientCache / Item ExpiresDate: Expire the content on the date of X Days from (Today). Only used when setting the Control property to use “UseExpires”.

#### IIS / Compression

<Compression>

<Node Action="Add" IISPath="" CompressionType="Static | Dynamic" State="Enabled | Disabled" />

</Compression>

* IIS / Compression / Node Action: Add. Add a compression scheme to the IISPath
* IIS / Compression / Node IISPath: IISPath to add the compression scheme to.
* IIS / Compression / Node CompressionType: Static | Dynamic. Sets the compression scheme to either static or dynamic.
* IIS / Compression / Node State: Enabled | Disabled

#### IIS / OutputCaching

<OutputCaching>

<Ext Action="" IISPath="" Extension="" KernelCaching="" UserCaching="" PeriodDurationSeconds="" Location="" />

<Ext Action="ClearAll" IISPath="IIS:\Sites\Padres" />

<Ext Action="Add" IISPath="IIS:\Sites\Chargers" Extension=".png" KernelCaching="CacheForTimePeriod" UserCaching="CacheUntilChange" PeriodDurationSeconds="9000" />

<Ext Action="Add" IISPath="IIS:\Sites\CleBrowns" Extension=".jpg" UserCaching="CacheUntilChange" />

</OutputCaching>

* IIS / OutputCaching / Ext Action: Add | ClearAll. Add an item to cache or REMOVE all items.
* IIS / OutputCaching / Ext IISPath: IIS Path to perform the action on.
* IIS / OutputCaching / Ext Extension: the file extension to cache
* IIS / OutputCaching / Ext KernelCaching: DontCache | CacheUntilChange | CacheForTimePeriod | DisableCache. Use Kernel-Mode Caching for this directive.
* IIS / OutputCaching / Ext UseCaching: DontCache | CacheUntilChange | CacheForTimePeriod | DisableCache Use User-Mode Caching for this directive.
* IIS / OutputCaching / Ext PeriodDurationSeconds: Duration in seconds to cache content. Used only when caching mode uses “CacheForTimePeriod”
* IIS / OutputCaching / Ext Location: Any | Client | Downstream | Server | None | ServerAndClient. Location to cache the content.

#### IIS / Config

IIS.Config allows you to specify IIS settings in the machine.config file <system.net> and <system.web> sections.

Action = Add | Set | Del. When add is set then the specified element will have the listed attributes added to it. The element itself will be added if it does not exist.

When the action is "Set" then the element will be replaced (or added if it doesn't exist) with the listed attributes.

When the action is "Del" then the element will be removed from the specification.

<Config>

<Setting Action="Set" Element="Add" Address="\*" maxConnection="100" />

<Setting Action="Set" Element="processModel" maxWorkerThreads="100" maxIoThreads="100" />

<Setting Action="Set" Element="httpRuntime" minFreeThreads="704" minLocalRequestFreeThreads="608" />

</Config>

### MSDTC

<MSDTC>

<NetworkDTC Action="Enabled" />

<XATransaction Action="Enabled" />

<SNALU62Trans Action="Enabled" />

<RemoteClients Action="Enabled" />

<RemoteAdmin Action="Enabled" />

<AllowInboundCom Action="Enabled" />

<AllowOutboundCom Action="Enabled" />

<MutualAuthReq Action="Ignore" />

<IncomingAuthReq Action="Ignore" />

<NoAuthReq Action="Enabled" />

<RestartAfterConf Action="Enabled" />

<LocalComputer>

<TransactionTimeout Value=:int16: />

<DCOMEnabled Value="True | False" />

<CISEnabled value="True | False" />

<DefaultAuthenticationLevel Value="Default | None | Connect | Call | Packet | Integrity | Privacy" />

<DefaultImpersonationLevel Value="Anonymous | Identify | Impersonate | Delegate" />

<RestartAfter Action="Enabled | Disabled" />

</LocalComputer>

<Perms>

<Perm RegKey="DefaultAccessPermission | DefaultLaunchPermission | MachineAccessPermission | MachineLaunchRestriction" ACE=:acespec:

</Perms>

</MSDTC>

\* Note when configuring MSDTC, the entire XML node needs to present but not the LocalComputer or Perms sub-elements.

* MSDTC / NetworkDTC: Enabled | Disabled | Ignore. Configure Network DTC Access
* MSDTC / XATransaction: Enabled | Disabled | Ignore. Configure XA Transactions
* MSDTC / SNALU62Trans: Enabled | Disabled | Ignore. Configure SNA LU 6.2 Transactions
* MSDTC / RemoteClients: Enabled | Disabled | Ignore. Allow Remote Clients. Must set Network DTC access to “Enabled”.
* MSDTC / RemoteAdmin: Enabled | Disabled | Ignore. Allow Remote Administration. Must set Network DTC access to “Enabled”.
* MSDTC / AllowInboundCom: Enabled | Disabled | Ignore. Allow Inbound Communication. Must set Network DTC access to “Enabled”.
* MSDTC / AllowOutboundCom: Enabled | Disabled | Ignore. Allow Outbound Communication. Must set Network DTC access to “Enabled”.
* MSDTC / MutualAuthReq: Enabled | Disabled | Ignore. Mutual Authentication Required. Must set Network DTC access to “Enabled”.
* MSDTC / IncomingAuthReq: Enabled | Disabled | Ignore. Incoming Caller Authentication Required. Must set Network DTC access to “Enabled”.
* MSDTC / NoAuthReq: Enabled | Disabled | Ignore. No Authentication Required. Must set Network DTC access to “Enabled”.
* MSDTC / RestartAfterConf: Enabled | Disabled | Ignore. Restart MSDTC service after configuration is completed.

### ServiceConfig

<ServiceConfig>

<Service Name="WAS" StartMode="Automatic" User="" />

<Service Name="NetMsmqActivator" StartMode="Automatic" User="True">

<Auth User="SVC\User" Password="\*\*\*\*\*\*\*\*\*\*" />

</Service>

</ServiceConfig>

* ServiceConfig / Service Name: Name of the service configure
* ServiceConfig / Service StartMode: Automatic | Manual | Disabled
* ServiceConfig / Service User: True | False. True to set up a user.
* ServiceConfig / Service /Auth User: Domain\User to run the service as. Used when ServiceConfig / Service User is set to True.
* ServiceConfig / Service / Auth Password: Password for the User. Used when ServiceConfig / Service User is set to True.
* ServiceConfig / ServicePerm / Action: Action to Log On As a Service (e.g. Remove)
* ServiceConfig / ServicePerm Name: the Account to action on for Log On As A Service

### Reg

<reg>

<Subkey Action="" Key="" Name="" Value="" type="" />

</reg>

* Reg / Action: Add | Delete | New
* Reg / Key: The Registry key location / Parent Path
* Reg / Name: The actual key name
* Reg / Value: the value of the “key name” (Only required for Action =”Add”)
* Reg / Type: The Value type (Only required for Action = “Add”)

### MSMQQueue

<MSMQQueue>

<Queue Action="Add" Name="prospectnotificationmanager/prospectdncmanager.svc" Public="" Transactional="true" Authenticated="" EncryptionReq="" EnableJournal="true">

<MaxJournalSizeKB></MaxJournalSizeKB>

<MaxQueueSizeKB> </MaxQueueSizeKB>

<Permissions>

<Perm Queue="" Public="" User="BRIDGEPOINT\dev\_svc\_prospect" ACE="Allow" Right="FullControl" Password=”Password”/>

</Permissions>

</Queue>

<AllQueues>

<Permissions>

<Perm User="BRIDGEPOINT\SCM" ACE="Allow" Right="FullControl" />

</Permissions>

</AllQueues>

<System>

<Permissions>

<Perm User="BRIDGEPOINT\DEV\_MSMQ\_FULL\_SG" ACE="Allow" Right="RecieveDeadLetter" />

<Perm User="BRIDGEPOINT\DEV\_MSMQ\_FULL\_SG" ACE="Allow" Right="PeekDeadLetter" />

</Permissions>

</System>

</MSMQQueue>

Note: Most elements are at the individual Queue level; however, permissions can be set at either the Queue or System level

* MSMQQueue / Queue Action: Add | Delete
* MSMQQueue / Queue Name: Queue Name to Add or Delete.
* MSMQQueue / Queue Public: True | False. Indicate if the queue to create/delete is a Public Queue. Defaults to false unless specified as true. Optional
* MSMQQueue / Queue Transactional: True | False. Indicate if the queue to create is transactional. Defaults to false unless specified as true. Optional
* MSMQQueue / Queue Authenticated: True | False. Indicate if the queue to create is an Authenticated queue. Defaults to false unless specified as true. Optional
* MSMQQueue / Queue EncryptionReq: None | Body | Optional. Indicate if the Encryption is required, and if so which one. The privacy level for the queue determines whether it accepts messages that are encrypted ("Body"), unencrypted ("None") or both ("Optional"). Defaults to “not encrypted” if no value is passed. Optional.
* MSMQQueue / Queue EnableJournal: True | False. Set Journaling on the specified to queue. Defaults to false if no value is passed. Optional.
* MSMQQueue / Queue / MaxJournalSizeKB: Size in KB to set the Maximum Journal Size. Automatically Enabled Journaling on this queue. Optional.
* MSMQQueue / Queue / MaxQueueSizeKB: Size in KB to set the Maximum Queue Size. Optional
* MSMQQueue / Queue / Permissions / Perm Queue: Name of the queue to adjust permissions on. Optional if creating a new queue and this element is nested.
* MSMQQueue / Queue / Permissions / Perm Public: True | False. Indicate if the queue is public (true) or private (false) Optional if creating a new queue and this element is nested.
* MSMQQueue / [Queue | System] / Permissions / Perm User: User to Adjust permissions for. Domain\UserName.
* MSMQQueue / Queue / Permissions / Perm Password: Password for the User
  + Used only in conjunction with “MSMQQueue / Queue EncryptionReq”
  + Will create the MSMQ Certificate (Enable-MSMQCertificate) for that user on the system being executed on
* MSMQQueue / [Queue | System] / Permissions / Perm ACE: Allow | Set | Deny | Revoke
  + Allow: Allow the defined “right” for that queue to the defined “User”.
  + Set: Set the access for “user” to be that absolute “right” for the queue.
  + Deny: Deny the “user” the “right” on the queue.
  + Revoke: Revoke that “right” from the “user” on the queue.
* MSMQQueue / [Queue | System] / Permissions / Perm Right: DeleteMessage | PeekMessage | WriteMessage | DeleteJournalMessage | SetQueueProperties | GetQueueProperties | DeleteQueue | GetQueuePermissions | ChangeQueuePermissions | TakeQueueOwnership | ReceiveMessage | ReceiveJournalMessage | GenericRead | GenericWrite | FullControl
  + DeleteMessage: Delete Message right
  + PeekMessage: Peek Message “right”.
  + WriteMessage: Write Message “right”.
  + DeleteJournalMessage: Delete Journal Message “right”.
  + SetQueueProperties: Set Queue “right”.
  + GetQueueProperties: Get Queue “right”.
  + DeleteQueue : Delete Queue “right”.
  + GetQueuePermissions: Get Queue Permissions “right”.
  + ChangeQueuePermissions: Change Queue Permissions “right”.
  + TakeQueueOwnership : Take Queue Ownership “right”.
  + ReceiveMessage: Receive Message “right”.
  + ReceiveJournalMessage: Receive Journal Message “right”.
  + GenericRead : Generic Read “right”.
  + GenericWrite : Generic Write “right”.
  + FullControl: Full Control “right”.
* MSMQQueue / AllQueues / Permissions / Perm User: Domain\User” to set permissions for on ALL Queues (on this machine)
* MSMQQueue / AllQueues / Permissions / Perm ACE: Allow | Set | Deny | Revoke. Sets an ACE for ALL Queues.
* MSMQQueue / AllQueues / Permissions / Perm Right: DeleteMessage | PeekMessage | WriteMessage | DeleteJournalMessage | SetQueueProperties | GetQueueProperties | DeleteQueue | GetQueuePermissions | ChangeQueuePermissions | TakeQueueOwnership | ReceiveMessage | ReceiveJournalMessage | GenericRead | GenericWrite | FullControl. Sets a Right for ALL Queues.

### Volume

<Volume>

<Part Action="Relocate" Label="DB" CurrentLetter="" NewLetter="Z:" />

<CDROM Action="MoveFirst" NewLetter="NextAvailable" MoveIf="D:" />

<iSCSI>

<Portal Address="10.2.16.31">

<LUN Action="Add" iQN="iqn.2011-03.org.example.istgt:d1-t2" Persistant="true" Label="DTC" DriveLetter="M:" />

</iSCSI>

<DirectAttached>

<FormatAndPartitionDisks>True</FormatAndPartitionDisks>

<Alignment>1024</Alignment>

<BlockSize>64k</BlockSize>

</DirectAttached>

</Volume>

* Volume / Part Action: Relocate | MoveFirst
* Volume / Part Label: Label of the existing partition in which to move or relocate.
* Volume / Part CurrentLetter: Current drive letter to move or relocate.
* Volume / Part NewLetter: DriveLetter: | NextAvailable. New Letter to move the volume to.
* Volume / CDROM Action: MoveFirst. Ensure that a CDRom drive is moved first, allways.
* Volume / CDROM NewLetter: DriveLetter: | NextAvailable. New Letter to move the volume to.
* Volume / CDROM MoveIf: DriveLetter: .Only move this CDROM if the drive letter matches.
* Volume / iSCSI /Portal Address: Address for an iSCSI target portal.
* Volume / iSCSI / LUN Action: Add. Add the lun to the machine.
* Volume / iSCSI / LUN iQN: iQN of the LUN to add
* Volume / iSCSI / LUN Persistant: True. Ensure that the iSCSI lun is marked as persistent. Defaults to false if no value is passed. Optional
* Volume / iSCSI / LUN Label. When formatting and partitioning disk, this is the label that will be set on this LUN.
* Volume / iSCSI / LUN DriveLetter. When formatting and partitioning disk, this is the drive letter that will be set on this LUN.
* Volume / DirectAttached / FormatAndPartitionDisks: When set to “True” any HDD disks that are attached to the computer, and currently do not have any partitions will be formatted. Disks will be labeled as “Disk#” where # is the number of the disk on the bus. (Disk1, Disk2). The partition will not have a drive letter assigned. Use Volume / Part / Action=”Relocate” to give the disk a drive letter.
* Volume / DirectAttached / Alignment: Use this to set the Disk alignment for all disks formatted and partitioned. Only used when FormatAndPartitionDisks is true.
* Volume / DirectAttached / BlockSize: Use this to set the block size for all disks formatted and partitioned. Only used when FormatAndPartitionDisks is true.

### EVLogs

<EVLogs>

<Log Action="" LogName="" EventSource="" Path="" LogRegKey="” />

</EVLogs>

* EVLogs / Log Action: Add | Move. Add a new log file, or move a log file physical location.
* EVLogs / Log LogName: LogFile Name.
* EVLogs / Log EventSource: Register an Event Source to log to this Log file.
* EVLogs / Log Path: Physical Path of the log file, or where you want to move the log file to.
* EVLogs / Log LogRegKey: Registry key location of the event log. Mandatory only during a move operation.

### Removefile

<RemoveFile>

<File Path="$TempPath" DateTime="(Get-Date)" DeletePathIfEmpty="False" Criteria="CreatedBefore" />

<File Path="C:\badpath" DateTime="((Get-Date).AddDays(-365)) " DeletePathIfEmpty="False" Criteria="CreatedBefore" />

</RemoveFile>

* RemoveFile / File Path: Path to the directory to clean files
* RemoveFile / DateTime: The string representing the DateTime object of files to be removed
* RemoveFile / DeletePathIfEmpty: remove the main path if completely empty of files post cleanup
* RemoveFile / Criteria: CreatedBefore | NotModifiedAfter. CreatedBefore - remove all files in the given Path that were *created before* the given date. NotModifiedAfter - remove all files in the given Path that have *not been modified after* the given date

### Filesystem

<Filesystem>

<PermInheritance>

<!--<Object Target="C:\Temp\MyFolder" AllowInheritance="True" PreserveInheritance="True" />-->

<!--<Object Target="" AllowInheritance="True | False" PreserveInheritance="True | False" />-->

</PermInheritance>

<Permissions>

<!--<Perm Action="DeleteUser" User="Bridgepoint\svc\_userA" Target="D:\Scripts\MyScript.ps1" Right="" Propagation="" Ace="" />-->

<!--<Perm Action="Add" User="BRIDGEPOINT\svc\_userB" Target="C:\WebForms" Right="CreateFiles" Propagation="None" Ace="Allow" />-->

<!--<Perm Action="Add" User="BRIDGEPOINT\svc\_userC" Target="C:\WebForms" Right="FullControl" Propagation="InheritOnly" Ace="Deny" />-->

<!--<Perm Action="Remove" User="BRIDGEPOINT\svc\_userD" Target="D:\MonkeyPictures" Right="ListDirectory" Propagation="NoPropagateInherit" Ace="Allow" />-->

<!--<Perm Action="Add | Remove | DeleteUser" User="DOMAIN\User" Target="C:\Files\whatever" Right="ReadData | ListDirectory | WriteData | CreateFiles | AppendData | CreateDirectories | ReadExtendedAttributes | WriteExtendedAttributes | ExecuteFile | Traverse | DeleteSubdirectoriesAndFiles | ReadAttributes | WriteAttributes | Delete | ReadPermissions | ChangePermissions | TakeOwnership | Synchronize | FullControl | Read | ReadAndExecute | Write | Modify" Propagation="None | NoPropagateInherit | InheritOnly" Ace="Allow | Deny" />-->

<!--<Perm Action="" User="" Target="" Right="" Propagation="" Ace="" />-->

</Permissions>

<Folders>

<!--<Item Action="Add" Path="C:\Path\to\Folder" />-->

<!--<Item Action="Delete" Path="C:\Path\to\Folder" />-->

</Folders>

<File Action="Link" Source="C:\udeploy-server\var\email" Destination="\\10.13.0.206\uDeploy\_repository\ucd\_test\_cluster\var\email" />

</Filesystem>

* Filesystem / PermInheritance /Object Target: Path to the Object on which to adjust NTFS Inheritance.
* Filesystem / PermInheritance /Object AllowInheritance: True | False. true to protect the access rules associated with this object from inheritance; false to allow inheritance.
* Filesystem / PermInheritance /Object PreserveInheritance: True | False. true to preserve inherited access rules; false to remove inherited access rules. This parameter is ignored if AllowInheritance is false.
* Filesystem / Permissions / Perm Action: Add | Remove | DeleteUser.
  + Add: Add a new permission to this ACL
  + Remove: Remove a permission from this ACL
  + DeleteUser: Purge all permissions for a user on this ACL
* Filesystem / Permissions / Perm User: User to take action on.
* Filesystem / Permissions / Perm Target: Filesystem object to adjust ACL’s on.
* Filesystem / Permissions / Perm Right: ReadData | ListDirectory | WriteData | CreateFiles | AppendData | CreateDirectories | ReadExtendedAttributes | WriteExtendedAttributes | ExecuteFile | Traverse | DeleteSubdirectoriesAndFiles | ReadAttributes | WriteAttributes | Delete | ReadPermissions | ChangePermissions | TakeOwnership | Synchronize | FullControl | Read | ReadAndExecute | Write | Modify
* Filesystem / Permissions / Perm Propagation: None | NoPropagateInherit | InheritOnly
* Filesystem / Permissions / Perm Ace: None | NoPropagateInherit | InheritOnly . Specifies how Access Control Entries (ACEs) are propagated to child objects.
  + None: Specifies that no inheritance flags are set.
  + NoPropagateInherit: Specifies that the ACE is not propagated to child objects.
  + InheritOnly: Specifies that the ACE is propagated only to child objects. This includes both container and leaf child objects.
* Filesystem / Folder / Item Action: Add | Delete. Add or Delete a folder.
* Filesystem / Folder / Item Path: Path to the folder to Add or Delete.
* Filesystem / File / Action: Link. Create a symbolic (soft) link to a file or folder

### AppConfig

<AppConfig>

<Octopus>

<Config Action="Add" AgentFolder="c:\Octopus Tentacle\agent" AppDir="C:\Octopus\Applications" Port="10933" TrustKey="8343CEC45768BC365ED632A73CB2E99328637EC3" TempCertDir="C:\temp" />

</Octopus>

</AppConfig>

* AppConfig / Octopus / Config Action: Add
* AppConfig / Octopus / Config AgentFolder: The octopus agent working folder.
* AppConfig / Octopus / Config AppDir: Directory to install Octopus.
* AppConfig / Octopus / Config Port: Octopus connection port.
* AppConfig / Octopus / Config TrustKey: Octopus server authentication key.
* AppConfig / Octopus / Config TempCertDir: Script working directory.

### WindowsFirewall

<WindowsFirewall>

<Form State="Enable | Disable" Profile=" " />

</WindowsFirewall>

* WindowsFirewall / Form State: Enable | Disable. Enable or Disable the windows firewall
* WindowsFirewall / Form Profile: allprofiles | currentprofile | domainprofile | global | privateprofile | publicprofile. Windows firewall profiles.

### MSSQL

<MSSQL>

<Install>

<ININame>DeveloperEdition.ini</ININame>

<EXEName>setup.exe</EXEName>

<Argument>/Q</Argument>

<Argument>/SQLSVCPASSWORD="password123"</Argument>

<Argument>/PID="ABCD-12345-EFGHI-12345-ABCD"</Argument>

<Argument>/AGTSVCPASSWORD=" password123"</Argument>

<Argument>/SAPWD=" password123"</Argument>

<LocalFolderName>DeveloperEdition</LocalFolderName>

</Install>

</MSSQL>

* MSSQL / Install / ININame: Name of the INI File to use which contains all the SQL installer parameters.
* MSSQL / Install / ExeName: MS SQL installer exe name. Parameters will be passed to this value.
* MSSQL / Install / Argument: Any MSSQL command line arguments to pass. Ensure that with any users that are passed in the INI file have passwords passed as command Line arguments. Also note that any passwords passed here. Also note that any passwords given will show up in the scripts log file (as clear text). Parameters can be found here: <http://msdn.microsoft.com/en-us/library/ms144259%28v=sql.105%29.aspx>
* MSSQL / Install / LocalFolderName: This is the name of the folder which contains the SQL binaries and MSSQL / Install / ExeName. This folder should be copied over using “Bindependency / Folder”. The value used for this parameter should be the name of the folder being copied locally.

### SpecialConf

<SpecialConf>

<SolrSpecial>

<MasterSlave>Master</MasterSlave>

</SolrSpecial>

<ConstellationSpecial>

<VMDDUpdate>True</VMDDUpdate>

</ConstellationSpecial>

</SpecialConf>

* SpecialConf / SolrSpecial / MasterSlave: Master | Slave. Set the Solr installation to be either a Master server or a Slave Server.
* SpecialConf / ConstellationSpecial / VMDDUpdate: True. Ensure that the VMWare Display driver update is installed.

### SearchAndReplace

<SearchAndReplace FilePath=:pathspec: FileFilter=:filespec: FindPattern=:xmlencodedstr: >

<Item ReplaceText=:xmlencodedstr: />

</SearchAndReplace>

### TextInsert

<TextInsert StartString=:xmlencodedstr: [IncludeStartString="True"] SourceFile=:filespec: [EndString=:xmlencodedstr:] [IncludeEndString="true"] TargetFile=:filespec: />

### FailoverClustering

<FailoverClustering>

<Cluster Name="Dev-SlyClus3a" ClusterIP="10.2.16.34">

<Nodes>

<Node Action="Add" NodeName="Dev-SlyClus6" />

<Node Action="Join" NodeName="Dev-SlyClus7" />

<Node Action="Join" NodeName="Dev-SlyClus8" />

</Nodes>

<Disks>

<Disk Name="Quorum" DriveLetter="Q:" />

<Disk Name="MsgQueue" DriveLetter="I:" />

<Disk Name="DTC" DriveLetter="M:" />

<Disk Name="Object" DriveLetter="O:" />

</Disks>

</Cluster>

</FailoverClustering>

* FailoverClustering / Cluster Name: Name of the cluster to create
* FailoverClustering / Cluster ClusterIP: IP Address to assign to the cluster
* FailoverClustering / Cluster / Nodes /Node Action: Add | Join. Add creates the cluster on that node, then will later join other nodes to the cluster. Join prepares nodes to join the cluster.
* FailoverClustering / Cluster / Nodes /Node NodeName: individual names of the cluster nodes.
* FailoverClustering / Cluster / Disks / Disk Name: Cluster Disk Label name.
* FailoverClustering / Cluster / Disks / Disk DriveLetter: Cluster Disk drive letter.

\* It should be noted that in order to properly use failover clustering when extra nodes are joined, you need to run the script on the primary “add” node, any “join” nodes, then lastly the “add” node again.

### Desired State Configuration (DSC)

DSC is a new management platform in Windows PowerShell that enables deploying and managing configuration data for software services and managing the environment in which these services run.

Microsoft DSC Overview: <https://technet.microsoft.com/en-us/library/dn249912.aspx>

<DSC MOF="\\10.13.0.206\scratch\DML\Microsoft\Powershell\MOF\BaseUCDServer\localhost.mof" />

* DSC / MOF: The path to the MOF (Managed Object Format) file that DSC will execute

## Notes

Please contact the SCM team (SCM@bridgepointeducation.com) with any issues or questions.