# CISDSC v0.0.1

Brown University Computing & Information Services, Windows Team
Authored by Yujun Qin, *Systems Administration Intern*

May 4, 2017

# Contents

# Chapter 1

# Project Overview

## 1.1 Introduction

CISDSC is an internal tool for the Windows Team at Brown University Computing & Information Services.
Many thanks to Mike R, Mike D, Adam, Tony, Geoff and Robert for the help.

## 1.2 Usage

Unless you want to customize your installation, or make changes to the template, this part should be sufficient to guide you through the installation process. Suppose the DSC pull server is to be installed on DWINSERVERNAME.ad.brown.edu. Also, suppose DWINS1.ad.brown.edu and DWINS2.ad.brown.edu are the servers we want to make compliant.
First check the prerequisites.

(i) Make sure you have WMF 5 on both the server and the clients. To check WMF version, run
`$PSVersionTable.PSVersion | % Major`;

(ii) Make sure you are in the administrators group for each machine.

(iii) Make sure you can access `\\files`.

Now install the pull server. **You should run each script as administrator.**

1. Put CisDsc.ps1 in a local drive. You can find it at
   `\\files\dfs\CISWindows\Software\DSCBackup\Scripts\DSC`.
   In this example, we put it under C drive.

2. Open PowerShell as administrator and run
   `. C:\CisDsc.ps1`;
   `Install-CisDscPullServer -ComputerName DWINSERVERNAME`;
   By default it is installed at C:\DSC.

3. Check that the pull server is up and running by visiting `http://DWINSERVERNAME.ad.brown.edu:8080/psdscpullserver.svc`.

4. To check if DWINS1 and DWINS2 are compliant, run
   `Check-CisDscCompliance -ComputerName DWINS1, DWINS2`;
   The first time your run this command, it should prompt you to build the localhost.mof file first. Give it a valid credential and proceed.

5. To see reports of compliance state, go to `C:\DSC\Reports`. There should be an overall report as well as detailed reports for each server.

6. To force DWINS1 to be compliant, run
   ```
   Install-PsDscConfiguration -ComputerName DWINS1
       -PullServerName -DWINSERVERNAME;
   ```

Turn on `-Verbose` flag for debugging purposes.

## 1.3 Support

Tony is in charge of the standard build doc, which the template is based on. You can find it at the SharePoint site. For more information, email Tony: anthony_jaworsky [at] brown.edu.
For help with installing and updating DSC pull server, email Yujun: yujun_qin [at] brown.edu.

# Chapter 2

# Script Files

## 2.1 CisDsc.ps1

Hi

## 2.2 Template.ps1

This is the golden image.

## 2.3 PullServerConfiguration.ps1

This is the

## 2.4 LCM_HttpPull.ps1

Hi

# Chapter 3

# Issues & TODOs

## 3.1 Issues

(i) During installation, CisDsc.ps1 copies backup DSC modules from \\files, rather than using the standard installation method. This is because the command `Install-Module` does not work in CIS network. The error is "Unable to download from URI 'https://go.microsoft.com/fwlink/?LinkID=627338&clcid=0x4'". The Network Team was not able to identify the problem. It is very likely that GPO blocks the connection.

(ii) Credential encryption is not implemented. To justify the use of domain user account, note that DSC Local Configuration Manager by default runs as SYSTEM. It does not have access to SMB share files unless we provide it with domain credentials. This is necessary for `[File]BGInfo`.
**Were this project to be continued, this should be top priority.**
For instructions, visit https://blogs.msdn.microsoft.com/powershell/2014/01/31/want-to-secure-credentials-in-windows-powershell-desired-state-configuration/.

(iii) HTTPS is not implemented. This can be easily configured once you have a certificate, but it could complicate the logic in `Install-CisDscPullServer`. The best practice would be to configure manually, after the pull server is installed.

## 3.2 TODOs

(i) Once credential encryption issue is resolved, uncomment the following lines in `[File]BGInfo`.

```
#SourcePath = "\\files\dfs\CISWindows\Software\DSCBackup\BGInfo32-64"
#MatchSource = $true
#Recurse = $true
```

```
#Credential = $Credential
```

(ii) Change current implementation of WMF 5 check. Rather than attempting to install it remotely, which always fails, maybe notify a systems administrator.

(iii) Find out why `[Script]DEP` in the template does not work on Server 2012 machines.

(iv) Add IISCrypto settings in template.