

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

Windows Server
2012 R2



Hands-on lab

In this lab, you will explore the use of Windows PowerShell Desired State Configuration (DSC). Windows PowerShell Desired State Configuration is a new set of Windows PowerShell extensions that allow you to define a statement-based configuration without having to worry about the specific implementation details of that configuration. In this lab, you will explore some basic DSC configurations, and then implement a more complex configuration to satisfy the prerequisites for SharePoint 2013 Foundation.

This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright 2013 © Microsoft Corporation. All rights reserved.

Microsoft Hyper-V, Internet Explorer, Windows, Windows PowerShell, and Windows Server 2012 are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Introduction

Objectives

After completing this lab, you will be able to:

- Explore new features in Management.
- Explore the features of Desired State Configuration in Windows PowerShell.

Prerequisites

Before working on this lab, you must have:

- An understanding of networking concepts.
- Familiarity with Windows PowerShell.

Overview of the lab

In this lab, you will explore the use of Windows PowerShell Desired State Configuration (DSC). Windows PowerShell Desired State Configuration is a new set of Windows PowerShell extensions that allow you to define a **statement**-based configuration without having to worry about the specific implementation details of that configuration. In this lab, you will explore some basic DSC configurations, and then implement a more complex configuration to satisfy the prerequisites for SharePoint 2013 Foundation.

Virtual machine technology

This lab is completed using virtual machines that run on Windows Server 2012 Hyper-V technology. To log on to the virtual machines, press CTRL+ALT+END and enter your logon credentials.

Computers in this lab

This lab uses computers as described in the following table. Before you begin the lab, you must ensure that the virtual machines are started and then log on to the computers.

Computer	Role	Configuration
DC	Domain controller	Domain controller
Server1	Sample server that is managed	Virtualized Hyper-V server
Admin	Administration client	Windows 8.1 with RSAT.

◆ All user accounts in this lab use the password **Passw0rd!**

Note regarding pre-release software

Portions of this lab include software that is not yet released, and as such may still contain active or known issues. While every effort has been made to ensure this lab functions as written, unknown or unanticipated results may be encountered as a result of using pre-release software.

Note regarding user account control

Some steps in this lab may be subject to user account control. User account control is a technology which provides additional security to computers by requesting that users confirm actions that require administrative rights. Tasks that generate a user account control confirmation are denoted using a shield

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

icon. If you encounter a shield icon, confirm your action by selecting the appropriate button in the dialog box that is presented.

Note on activation

The virtual machines for these labs may have been built by using software that has not been activated. This is by design in the lab to prevent the redistribution of activated software. The unactivated state of software has been taken into account in the design of the lab. Consequently, the lab is in no way affected by this state. For operating systems other than Windows 8.1, click Cancel or Close if prompted by an activation dialog box. If you are prompted by an Activate screen for Windows 8.1, press the Windows key to display the Start screen.

Exercise 1: Exploring Windows PowerShell Desired State Configuration

In this exercise, you will explore some of the new enhancements in management for Windows Server 2012 R2. You will then explore a key new feature called Windows PowerShell Desired State Configuration (DCS). This feature allows you provide standardization of services across server farms and server deployments for specific roles within your organization. DCS makes use of different types of resources to extend the type of server functionality validation.

Reviewing Windows PowerShell Desired State Configuration components

In this task, you will review the different types of resources used by Windows PowerShell Desired State Configuration. The resources to use are presented as providers.

✎ Begin this task logged on to **Server1** as **Contoso\Administrator** using the password **Passw0rd!**

1. Using File Explorer, navigate to

C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PSDesiredStateConfiguration\PSProviders.

✎ The resources available are:

Resource	Purpose
Archive	Unpack archive files (.zip) at a path
Registry	Manage registry keys and values
Script	Ability to run Windows PowerShell script blocks
Package	Install and manage MSI packages
Environment	Manage system environment variables
Group	Manage local groups
User	Manage local user accounts
Service	Manage services
File	Manage files and folders
Log	Write messages to the Microsoft-Windows-Desired State Configuration even log
Process	Configure processes
Role	Add or remove Windows Roles and Features

Implementing basic DCS

In this task, you will use create a DCS script to deploy fully functional websites to an unconfigured Windows Server 2012 R2 server. You will also learn how DCS can be used to mitigate configuration drift.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the taskbar, right-click the **Windows PowerShell** icon, and then click **Windows PowerShell ISE**.
2. On the File menu, click **New Remote PowerShell Tab**.
3. In the New Remote PowerShell Tab window, in computer, type **Server1**, and then in the User name, type **Administrator**, and then click **Connect**.
4. Enter the password **Passw0rd!** when prompted, and then click **OK**.
5. On the View menu, click **Show Script Pane**.
6. In the Windows PowerShell ISE Script Pane, type the following commands, pressing ENTER after each one.

◇ The code for this script file can also be found in `c:\LabFiles\DSC-Scripts.ps1`. Instead of typing you can optionally copy the relevant code sections from that file.

★ You will be creating a script to add IIS and ASP.NET 4.5 to Server1.

```
Configuration IISWebsite
{
    Node Server1
    {
        WindowsFeature IIS
        {
            Ensure = "Present"
            Name = "Web-Server"
        }
        WindowsFeature ASP
        {
            Ensure = "Present"
            Name = "Web-Asp-Net45"
        }
    }
}
IISWebSite
```

7. Save the script in **C:\Labfiles** as **DSC-Server1-IIS.ps1**.

8. Press **F5** to run the file.

★ This basic state configuration file describes the desire to have a basic web server with ASP.NET 4.5 installed.

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

- ✦ The output of this configuration statement is a MOF file for the server Server1, in a folder named IISWebSite. The folder represents the configuration set, and the MOF file a specific NODE (Server) in that configuration set.

9. In Windows PowerShell ISE, type the following commands, pressing ENTER after each one.

- ✦ This will verify that the web server and ASP.NET are not installed, apply the desired state which will install the web server and ASP.NET, and then verify the results.

```
↪ Get-WindowsFeature -ComputerName Server1 -Name Web-ASP*
↪ Start-DSCConfiguration -ComputerName Server1 -Path IISWebSite -Wait
  -Verbose
↪ Get-WindowsFeature -ComputerName Server1 -Name Web-ASP*
```

- ✦ WEB-ASP-NET45 is now installed.

10. In Windows PowerShell ISE, type the following command, and then press ENTER.

```
↪ Get-Service -Name W3SVC -ComputerName Server1
```

- ✦ This will verify that the web server and website are now running,

Implementing more detailed DCS

In this task, you will expand the Windows PowerShell Desired State Configuration setup to include the configuration of a website, in addition to the basic installation requirement deployed in the previous tasks.

- ✍ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the taskbar, right-click **Windows PowerShell**, and then click **Windows PowerShell ISE**.
2. On the File menu, open a **New Remote PowerShell Tab**, and then connect to **Server1**.
3. In Windows PowerShell ISE, navigate to **C:\LabFiles\Server1**, and then open

DSC-Server1-IIS.ps1.

- ✦ This is the file you created in the previous tasks

4. In the Windows PowerShell ISE Script Pane, review the script.

```
↪ Configuration IISWebsite
↪ {
↪     Node Server1
↪     {
↪         WindowsFeature IIS
↪         {
↪             Ensure = "Present"
↪             Name = "Web-Server"
↪         }
↪     }
↪ }
```

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

```
↳ WindowsFeature ASP
↳ {
↳     Ensure = "Present"
↳     Name = "Web-Asp-Net45"
↳ }
↳ }
↳ }
↳ IISWebSite
```

5. Add the following commands on line 15 after the closing brace in the ASP WindowsFeature section.

- ✦ This script is a file block. It will copy the BakeryWebsite content from a source directory to the standard IIS folder path.

- ◇ The code for this script file can also be found in `c:\LabFiles\DSC-Scripts.ps1`. Instead of typing, you can copy the relevant code sections from that file.

```
↳ File WebContent
↳ {
↳     Ensure = "Present"
↳     Type = "Directory"
↳     SourcePath = "C:\labfiles\Source\BakeryWebsite"
↳     DestinationPath = "C:\inetpub\WWWRoot\"
↳     Recurse = $true
↳ }
```

6. The completed script:

```
↳ Configuration IISWebsite
↳ {
↳     Node Server1
↳     {
↳         WindowsFeature IIS
↳         {
↳             Ensure = "Present"
↳             Name = "Web-Server"
↳         }
↳         WindowsFeature ASP
↳         {
↳             Ensure = "Present"
↳             Name = "Web-Asp-Net45"
↳         }
↳         File WebContent
```


Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

```
↪      {  
↪          Ensure = "Present"  
↪          Type = "Directory"  
↪          SourcePath = "C:\labfiles\Source\BakeryWebsite"  
↪          DestinationPath = "C:\inetpub\WWWRoot\  
↪          Recurse = $true  
↪      }  
↪  }  
↪ }  
↪ IISWebSite
```

7. Press **F5** to run the file.

✦ This configuration set, in addition to ensuring the role for web server is present, will configure the initial website by copying the files from a distribution source. It will produce an updated MOF file.

8. Using Internet Explorer, navigate to **http://server1**.

✦ The default home page will be displayed.

9. In Windows PowerShell ISE, type the following command, and then press ENTER:

```
↪ Start-DSCConfiguration -ComputerName Server1 -Path IISWebSite -Wait  
-Verbose
```

10. In Internet Explorer, refresh the website **Server1**.

✦ The updated website content is displayed.

Implement repeatable configuration and change control

In this task, you will use DCS to recover from a change to a website which results in the website being down.

✍ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Windows PowerShell ISE, type the following command, and then press ENTER.

```
↪ REMOVE-ITEM \\Server1\c$\inetpub\wwwroot\images -Force
```

2. Click **Yes**.

3. Navigate to **http://server1**.

✦ Note that all images are missing.

4. In Windows PowerShell ISE, type the following command, and then press ENTER.

```
↪ Start-DSCConfiguration -ComputerName Server1 -Path IISWebSite -Wait  
-Verbose
```

5. In Internet Explorer, refresh the website **Server1**.

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

✦ The corrected content is displayed.

6. If you are going to be performing the next (optional) exercise, leave the Windows PowerShell ISE console open.

(Optional) Exercise 2: Using Windows PowerShell Desired State Configuration for SharePoint

In this exercise, you will implement a pre-created configuration script to prepare a server to receive a SharePoint 2013 installation. This exercise is an example of how DCS scripts can be created and distributed to simplify the process of server installation.

- ◆ The prerequisites for SharePoint foundation are extensive and take quite a bit of time to apply. Only complete this exercise if you are prepared to wait up to 15 minutes. This is due to extensive installation and the need to download some components from the Internet.

Implement repeatable configuration and change control

In this task, you will use DCS to recover from the removal of a key item in the registry needed for local access to the SharePoint site that will be deployed.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In the Windows PowerShell ISE console you left open from the previous exercise, open the file

C:\LabFiles\DSC-SharePoint.ps1.

- ✦ This file includes all role prerequisites needed for a SharePoint Foundation 2013 deployment.
- ✦ You may observe that the Media Foundation has been marked as commented, causing it to be skipped. Media Foundation requires a system restart so it was omitted from this configuration run simply to expedite the configuration processing.
- ✦ If you open a new Windows PowerShell ISE console, ensure that you connect to Server1 using the New Remote PowerShell tab from the File menu.

2. In the Windows PowerShell ISE Script Pane, review the script.

3. Press **F5** to run the file.

- ✦ This configuration set, in addition to ensuring web server role is present, will configure the initial website by copying the files from a distribution source. It will produce an updated MOF file.

4. In Windows PowerShell ISE, type the following command, and then press ENTER.

```
↪ Start-DSCConfiguration -ComputerName Server1 -Path SharePointPrereq  
-Wait -Verbose
```

- ◆ This command may take up to 15 minutes to complete.

5. In Windows PowerShell ISE, type the following command, and then press ENTER.

```
↪ Get-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\LSA  
-Name DisableLoopbackCheck
```

- ✦ The value and entry for DisableLoopbackCheck is presented.

Windows Server 2012 R2 – Windows PowerShell Desired State Configuration

✦ For this command to succeed, you must be connected to Server1 in the PowerShell ISE.

6. In Windows PowerShell ISE, type the following command, and then press ENTER:

```
↪ REMOVE-ITEMProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\LSA  
-Name DisableLoopbackCheck
```

✦ The registry key is removed.

7. In Windows PowerShell ISE, type the following command, and then press ENTER:

```
↪ Get-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\LSA  
-Name DisableLoopbackCheck
```

✦ The value and entry for DisableLoopbackCheck is no longer listed and the command returns an error.

8. In Windows PowerShell ISE, type the following command, and then press ENTER:

```
↪ Start-DSCConfiguration -ComputerName Server1 -Path SharePointPrereq  
-Wait -Verbose
```

9. In Windows PowerShell ISE, type the following command, and then press ENTER:

```
↪ Get-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\LSA  
-Name DisableLoopbackCheck
```

✦ The value and entry for DisableLoopbackCheck is presented and restored based on the standardization of the configuration for Server1.