

Guide for SharePoint PowerShell Content Deployment

Ayoub Umoru

Published: June 2013

Send feedback or suggestions about this document to Ayoub@Umoru.de or write a reply.

Contents

Guide for SharePoint PowerShell Content Deployment.....	4
Guide History.....	4
Supported Configurations	4
Prerequisites.....	4
Mandatory Configuration.....	4
Files in this Solution.....	5
SharePoint PowerShell Content Deployment Purpose	5
Content Deployment Scenarios	5
Configure Content Deployment between two distinct Farms. ..	Error! Bookmark not defined.
Execute Content Deployment between two distinct Farms.....	6
Exporting Content from Farm A	7
Importing Content from Farm A.....	7
How SharePoint Content Deployment works.....	7
Links	8
Appendix: PowerShell Cmdlets.....	8
Token	8
Import	9
Export.....	9

Guide for SharePoint PowerShell Content Deployment

This guide was written based on version 0.2 of the PowerShell Content Deployment Release

Guide History

Release Date	Changes
June 17, 2013	Original release of this guide

Supported Configurations

This SharePoint PowerShell Content Deployment solution needs a local SharePoint 2010 installation.

SharePoint 2010 – SharePoint 2010	Support
SharePoint 2013 – SharePoint 2013	Support
SharePoint 2010 – SharePoint 2013	Not tested
Different Patch Levels between 2 Farms	Support
Different Feature Sets between 2 Farms	Support
Disconnected Farms (Manual Data Transfer)	Support

Prerequisites

The SharePoint PowerShell Content Deployment has the following prerequisites:

- Installed Version of SharePoint 2010
- Enabled Windows PowerShell (Unsigned Scripts allowed)

Mandatory Configuration

Mandatory configuration steps are required and described [here in the document](#).

Files in this Solution

The SharePoint PowerShell Content Deployment Solution includes the following files:

- Console.bat
- Module.psm1
- Settings/Local/Settings.xml
- PSModules/Configuration.ps1
- PSModules/ContentDeployment.Import.ps1
- PSModules/ContentDeployment.Export.ps1

SharePoint PowerShell Content Deployment Purpose

The SharePoint PowerShell Content Deployment Solutions transfers content between 2 distinct SharePoint Farms.

- [Configure Content Deployment between two distinct Farms](#)
- [How SharePoint Content Deployment works](#)

For details on the discoveries, rules, monitors, views, and reports contained in this monitoring pack, see [Appendix: PowerShell Cmdlets](#).

Configure Content Deployment between two distinct Farms

This scenario describes the Steps involved for implementing Content Deployment between two Farms

1. Create a destination location/Shared directory witch is r/w accessible from both Farms
2. Get the latest Version from SVN
3. Locate the Settings.xml and edit the paths and directories.1

<Root>

<ContentDeployment>

<Export

FileLocation=\\SharedStorage\\ContentDeploymentStorage\\Data

TokenFileLocation=\\SharedStorage\\ContentDeploymentStorage\\Tokens\\Token.token>

4. Create the Job entry in Settings.xml describing on what you want to Export.
Use the section JobItems to Name every List that you want to transfer.

```
<Root><ContentDeployment><Export><Jobs>
<Job
  Name="XXX <- Your custom Job Name"
  SiteUrl="[SOURCEURL]"
  WebUrl="[SOURCEURL]"
  RetainObjectIdentity="0"
  Type="List"
  FileCompression="0"
  ExcludeDependencies="1"
  ImportUserInfoDateTimeOption="ImportAll"
  DestinationSiteUrl="[DEST URL]"
  DestinationWebUrl="[DEST URL]">
  <JobItems>
    <JobItem Name="Calendar" Type="List" />
    <JobItem Name="Shared Documents" Type="List" />
  </JobItems>
</Job>
```

5. Finish Editing the Settings.xml
6. Copy the Solution local to any Server on Farm A. (Correct paths to the Token Store and Data Store if necessary)
7. Copy the Solution local to any Server on Farm B. (Correct paths to the Token Store and Data Store if necessary)

Execute Content Deployment between two distinct Farms.

This scenario describes the Steps involved for executing Content Deployment between two Farms.

Exporting Content from Farm A

This scenario describes the Steps involved for Exporting Content from Farm A using the SharePoint PowerShell Content Deployment solution.

1. Run Console.bat

2. Type in and execute:

Run-ContentDeploymentJob –Identity XXX<- Your configured Job Identification –Mode Full

After an successful Full Export and an following import, you then can switch to Export Content incremental by type in and execute the following command:

Run-ContentDeploymentJob –Identity XXX<- Your configured Job Identification –Mode Incremental

Importing Content from Farm A

This scenario describes the Steps involved for Importing Content from Farm A using the SharePoint PowerShell Content Deployment solution.

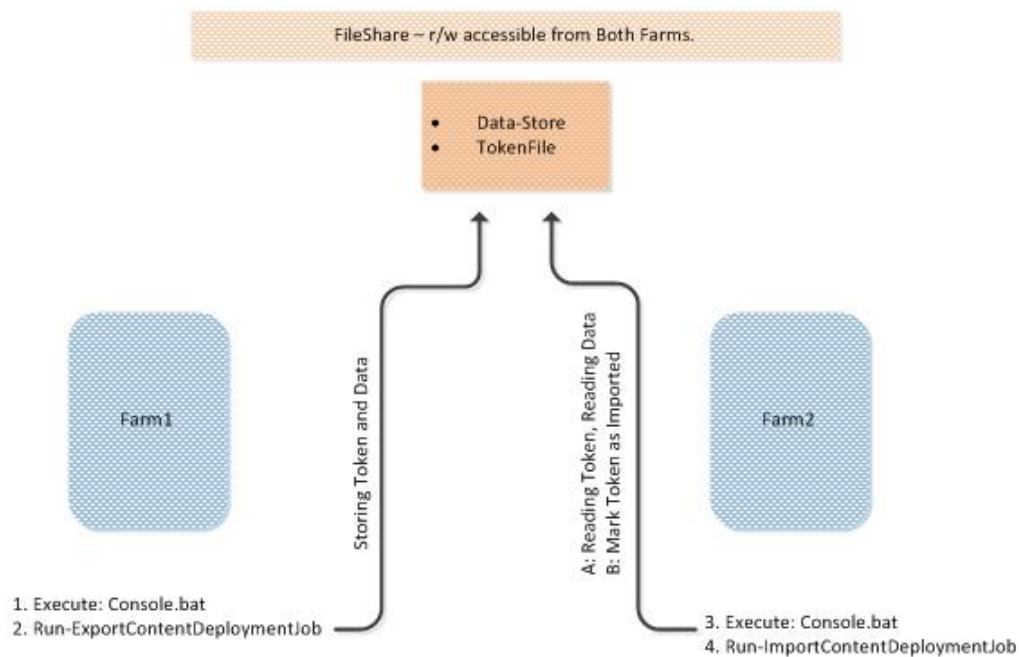
3. Run Console.bat

4. Type in and execute:

Run-ContentDeploymentJobForImport –Identity XXX<- Your configured Job Identification

How SharePoint Content Deployment works

The following diagram shows how the SharePoint PowerShell Content Deployment works.



Links

The following links connect you to information about common tasks that are associated with Content Deployment in General:

- [Stefan Grossner](http://blogs.technet.com/b/stefan_gossner/archive/tags/content+deployment/)
(http://blogs.technet.com/b/stefan_gossner/archive/tags/content+deployment/)

A useful resource is the TechNet Library provided by Microsoft.

- Microsoft.SharePoint.Deployment Namespace ([http://msdn.microsoft.com/en-us/library/ms427553\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/ms427553(v=office.14).aspx))

Appendix: PowerShell Cmdlets

The SharePoint PowerShell Content Deployment Solution contains a hand full Cmdlets for Test and Troubleshooting Purpose and also for executing the Main tasks. Export and Import (Bolt).

Token

Cmdlet	Parameter	Description
Get-TokenFile	-	Reads the token file and returns the token objects
Get-Token	-Identify XXX	Return the latest Token object for Job XXX, if marked as already imported or not
Get-TokenForJob	-Job XXX	Returns the latest unused Token object for Job

Import

Cmdlet	Parameter	Description
Run-ContentDeploymentJobForImport	-Identity XXX	Imports the content into SharePoint for Job XXX

Export

Cmdlet	Parameter	Description
Run-ContentDeploymentJob	-Identity XXX -Mode [Full/Incremental]	<p>General: Reads the configuration</p> <p>-Mode Full: Exports the content from SharePoint for Job XXX, creates a new entry in the Token-File and stores the package in the Target Location.</p> <p>-Mode Incremental: Checks if the last Import was successful to prevent data not being imported correctly (Accidentally skipping an incremental import)</p> <p>Exports the changed</p>

Cmdlet	Parameter	Description
		content from SharePoint for Job XXX, creates a new entry in the Token-File and stores them in the Target Location.
Get-ContentDeploymentJob	-Identity XXX	Gets the object of the Content Deployment Configuration for Job XXX
Get-ContentDeploymentJobChanges	-Identity XXX (Optional)–Object NameOfTheList	Returns objects for Job XXX that has been changed since the last Export operation.
Get-ContentDeploymentJobChangesDistinct	-Identity XXX	Returns distinct objects for Job XXX that has been Updated or Added since the last Export operation.