Scripted Server Farm Installation

Ben Curry, SharePoint Server 2007 MVP

March 4, 2008

© Copyright 2008, English, Bleeker & Associates, Inc., DBA Mindsharp (www.mindsharp.com).

This paper and it’s contents is copyrighted and owned by English, Bleeker & Associates, Inc. and is intended for the private use of those to whom it is given by English, Bleeker & Associates, Inc. No part of the paper may be reproduced in any format by any method without the express written consent of English, Bleeker & Associates, Inc. This paper may not be distributed or re-distributed in any form without the express written consent of English, Bleeker & Associates.

English, Bleeker & Associates, Inc. makes no guarantees or warranties, express or implied, about the contents and/or distribution of this paper. English, Bleeker & Associates, Inc. is not responsible for any adverse or unwanted effects or results that might arise from using the contents or advice of this paper and we reserve the right to change the contents of this paper at any time without notice. The information in this paper is provided “: AS IS”.

Mindsharp® is a registered trademark of English, Bleeker & Associates, Inc. in the United States.

Microsoft®, SharePoint®, Office®, and Windows® are registered trademarks of the Microsoft Corporation in the United States and in other Countries.

Nothing in this paper should be construed as legal advice in any way nor should the contents of this paper be relied upon to replace legal advice from a licensed, practicing attorney.

Mindsharp does not warrant the contents of this paper. Information in this paper and the contents of this paper are subject to change without notice.

Table of Contents

[Introduction 3](#_Toc192241588)

[Modifying Config.xml and Installing the Binaries 3](#_Toc192241589)

[Provisioning Services with PSConfig.exe 6](#_Toc192241590)

[Installing on the First Server in the Farm 6](#_Toc192241591)

[Installing on Subsequent Servers in the Farm 6](#_Toc192241592)

[Lessons Learned 6](#_Toc192241593)

# Introduction

There are many reasons to automate a SharePoint Server 2007 server farm install, but consistency and farm recovery are the two primary. You could, arguably, build a farm just as quickly with a manual install. The point isn’t really to ‘build a farm faster’, it is to build a farm unattended and consistently. If you are deploying many server farms, then using an unattended script will allow you to do this more easily when remote. It will also allow a very similar farm setup throughout your enterprise.

This whitepaper will be more about using scripted farm installs for the purpose of farm recovery, but can be used for many purposes. One of the keys to scripting a farm install is attention to detail. In fact, if you script your farm install for the purpose of disaster recovery and don’t get *exactly* the results you want, then you must uninstall and start over. Why? Because I have learned that changing the script to what the desired result is, without testing, doesn’t always work! There are so many dependencies of SharePoint Server 2007 that you must have a perfectly running script the first time. Additionally, you must modify your script every time you change the farm topology, with content databases and Web applications being the primary script edits. If your scripts were already wrong in the beginning, then you only increase your likelihood of failure when future edits are required.

This whitepaper is not meant to be all-inclusive of every nuance when scripting a farm install. Instead, it is a compilation of what we have learned scripting farm installs for the purpose of disaster recovery. We were not happy with the native tools for restoring a farm from total failure, and did not want to rely on third-party tools for the job. Our solution was to completely script the farm installation so that we could rebuild it quickly. We can then restore the databases from SQL Server backups, and only rely on the native SharePoint Server 2007 tools for Shared Services Providers recovery.

Note that the attached scripts have not been re-factored using variables because every farm installation is very unique. You will need to edit for your specific implementation. The attached scripts were built for a SQL Server named *app1*, an Index/Central Administration server names *app2*, and two WFE Servers running the query role named *wfe1* and *wfe2.*

# Modifying Config.xml and Installing the Binaries

To begin scripting a SharePoint Server 2007 server farm, you need to create a copy of the config.xml file that comes on the installation media. You can find several examples on the installation media under x86\Files\Setup<type> for 32 bit media, and x64\Files\Setup<type> for 64 bit media. Microsoft recommends that you do NOT use an XML editor, such as Visual Studio, to edit this file. They recommend you use notepad.exe so the formatting isn’t modified.

**Tip:** You can use the config.xml included with this whitepaper package. It is for a farm installation to include SharePoint Server 2007, and will not ask for input. If you do not want to show the display, simply change <Display Level = “None”>.

You can find a constantly updated reference guide for config.xml at *http://technet2.microsoft.com/Office/en-us/library/003e5316-1018-4949-ac54-42fa12c5e51e1033.mspx*. The relevant settings for this whitepaper will be included here for convenience. First, you need to decide which binaries and services you will install. This is very important. You *must* install both ‘sts’ and ‘spswfe’ for Windows SharePoint Services 3.0 and SharePoint Server 2007, respectively. If you do not install ‘spswfe’, you will be left with a Windows SharePoint Services 3.0-only installation. The following xml will install both Windows SharePoint Services 3.0 and SharePoint Server 2007, with all server roles.

<Package Id="sts">

<Setting Id="LAUNCHEDFROMSETUPSTS" Value="Yes"/>

<Setting Id="REBOOT" Value="ReallySuppress"/>

<Setting Id="SETUPTYPE" Value="CLEAN\_INSTALL"/>

</Package>

<Package Id="spswfe">

<Setting Id="SETUPCALLED" Value="1"/>

<Setting Id="REBOOT" Value="ReallySuppress"/>

<Setting Id="OFFICESERVERPREMIUM" Value="1" />

</Package>

Don’t be fooled by the *wfe* in ‘spswfe’. It does not affect the installation type (Web Front-end, Application, Basic). You will define the installation type later using ‘SERVERROLE’. After defining both Windows SharePoint Services 3.0 and SharePoint Server 2007 to be installed, you should define the installation directory for the binaries. This will not change where the *12 Hive* is located, only where *Microsoft Office Servers* program files are located. For what’s its worth, we generally install into the default %program file% location. If you do not define the installation location, it will install into %program files%\Microsoft Office Servers. Note that the config.xml uses system environment variables.

You will want robust logging if installing remotely, so be careful to define a logging location you can see from your remote location. The default logging location is in %temp%. Note that this is for the binary installs only, not for farm provisioning. Farm provisioning logs are in the *12 Hive\Logs* directory. To set the binary installation logs, you can change/add the following xml to config.xml:

<INSTALLLOCATION Value="c:\Program Files\Microsoft SharePoint" />

After defining the binaries location, you need to decide how you will display progress and enter the Product ID (Product Key). Depending on how your licensing is arranged with Microsoft, this key may need to be altered for every server. In the following example, we will be able to see progress of the installation, but setup.exe will not require any console input. And no, the key won’t work for you ☺

<Display Level="basic" CompletionNotice="no" AcceptEULA="Yes" />

<PIDKEY Value="F6YVR-4XY7K-RITR4-N0T-A-CHANCE" />

There are x options for Display Level: *none* will show no UI at all. You must check taskmgr.exe for the *setup.exe* process to finish. *Basic* will show the UI screen, but will not ask for user input. *Verbose* will present you with the standard UI for your server installation type.

Lastly, but definitely NOT least, is the server installation type. The server installation type defines if it will be *standalone, Web Front-end, or Complete*. Note that you cannot use the same verbiage you see in the UI. The following are the acceptable *SERVERROLE* VALUES:

* APPLICATION
* SINGLESERVER
* WFE

Unless you want SQL Server Embedded Edition, do not select SINGLESERVER. Likewise, we generally install all servers with the APPLICATION server type. Even if you only need to render Web content on this server, you can use it in the event another application server fails. Be aware that most WFE Servers in a medium and larger farm have the query role on all WFE Servers. If you install in ‘WFE’ mode, you will not have the search service available, and therefore the query role. Finally, install the binaries using *%dvdloc%\x86\setup /config c:\scripts\config.xml*.

# Provisioning Services with PSConfig.exe

After your binaries are installed, you can continue provisioning the server farm. If you did not choose to display a UI status, be sure *setup.exe* has finished by viewing current processes before continuing. Remember, a server ‘farm’ = configuration database. You can build an entire server farm with only one SharePoint Server 2007 system. We will not cover all pre-requisites on building a server farm in this whitepaper, those are available at *http://technet.microsoft.com/office/sharepointserver/default.aspx.* We will focus on using psconfig.exe and stsadm.exe to provision a server farm.

First, you must create the server farm – the configuration database. After you installed the binaries, you have access to psconfig.exe in the 12 Hive, more specifically, *%Program Files%\Common Files\Microsoft Shared\web server extensions\12\BIN\psconfig.exe*. You will notice the scripts attached to this whitepaper have the full path included as a best practice. To create the server farm, define the SQL Server instance and database name, as follows:

psconfig.exe -cmd configdb -create -server app1 -database SharePoint\_Config\_Contoso -user contoso\mossfarm -password P@ssw0rd -admincontentdatabase Central\_Admin\_Content

The above command created a server farm on a SQL Server instance named *app1*, with a configuration database named *SharePoint\_Config\_Contoso*. It also created a content database and embedded site collection for Central Administration. This occurs when using the GUI – you just don’t see it. It is important to understand that the configuration database and Central Administration are tightly coupled, you might say they are even married. You cannot backup one without the other, and you restore one without the other. Note that we are using the contoso\mossfarm account as the installer account. This account only needs administrative permission during installation, and you can remove its local administrator permissions after installation is complete.

After the server farm is created, you need to provision Central Administration on at least one server in the farm. Yes, you have created in SQL Server, but you must still provision an IIS server to render the content. For this example, we will install on *app2*, the first server in the farm

psconfig.exe -cmd adminvs -provision -port 12345 -windowsauthprovider onlyusentlm

The attached scripts assume you are installing the Index server into the farm first. There are two primary reasons for doing so. First, it allows you to completely build your farm and index content sources, without moving the Index server role at a later day. Second, we commonly provision Central Administration on the Index Server. After provisioning Central Administration, you need to install all services associated with your installed server role. We installed an APPLICATION server, so we should get all services installed.

psconfig.exe -cmd services install

Next, and VERY important, is setting the security on registry settings and the file system. If you do not do so, your installations will complete correctly, but you will have miscellaneous errors until the end of time, remove the installation, or until you simply run the command to fix your farm.

psconfig.exe -cmd secureresources

After you have created the farm, provisioned Central Administration, installed services, and secured resources, you need to start required services, such as Search. The following sections are optional, but the most commonly enabled services. First, we will start Search with the following stsadm.exe command, which can be found along side psconfig.exe in the 12 Hive\Bin directory:

stsadm.exe -o osearch -action start -role Index -farmcontactemail administrator@contoso.msft -farmperformancelevel maximum -farmserviceaccount "contoso\mossservice" -farmservicepassword P@ssw0rd

**Tip:** You can download an stsadm.exe whitepaper at Mindsharp’s Premium content are for free. It includes the *osearch* commands and all stsadm.exe command that are in this whitepaper

After starting the Search service, you can continue by starting Windows SharePoint Services 3.0 search, installing all features (important), and creating the Shared Services Provider administration Web application and My Sites Web application. The following commands are examples using *contoso.msft* as the domain and create a Shared Services Provider named *SSP*. These scripts can be found in the accompanying script files.

stsadm.exe -o spsearch -action start -farmserviceaccount contoso\mossservice -farmservicepassword P@ssw0rd -farmcontentaccessaccount contoso\mosscrawler -farmcontentaccesspassword P@ssw0rd -databaseserver app1 -databasename Contoso\_WSS\_App2\_Search

:: Install all features on machine

psconfig.exe" -cmd installfeatures

:: Create My Site Web application. Verify database name and administrator's names.

stsadm.exe -o extendvs -url http://my.contoso.msft -ownerlogin "contoso\administrator" -owneremail "administrator@contoso.msft" -exclusivelyusentlm -ownername "Administrator" -databaseserver app1 -databasename Contoso\_WSS\_Content\_MySites\_01 -sitetemplate spsmsitehost -description "My Site Host" -sethostheader -apidname MySiteAppPool -apidtype configurableid -apidlogin contoso\mysiteapid -apidpwd P@ssw0rd

:: Enable Self Service Site Management (Creation) on http://my.contoso.msft

stsadm.exe -o enablessc -url http://my.contoso.msft

:: Create SSP Web application. Verify database and apid names. (APID = Application Pool Identity)

stsadm.exe -o extendvs -url http://ssp.contoso.msft -exclusivelyusentlm -databaseserver app1 -databasename Contoso\_WSS\_Content\_SSP -donotcreatesite -description "SSP Admin Host" -sethostheader -apidname SSP -apidtype configurableid -apidlogin contoso\sspapid -apidpwd P@ssw0rd

:: We must reset IIS before building the SSP. If you are local on the box, you can check all services are created before creating SSP.

stsadm.exe -o createssp -title "Contoso SSP" -url http://ssp.contoso.msft -mysiteurl http://my.contoso.msft -ssplogin contoso\sspservice -indexserver app2 -indexlocation c:\indexes -ssppassword P@ssw0rd -sspdatabaseserver app1 -sspdatabasename Contoso\_SSP\_Config -searchdatabaseserver app1 -searchdatabasename Contoso\_SSP\_Search -ssl no

:: Creating Portal. Change the URL if you will have a different URL than 'portal'.

stsadm.exe -o extendvs -url http://portal.contoso.msft -ownerlogin "contoso\administrator" -owneremail "administrator@contoso.msft" -exclusivelyusentlm -ownername "Administrator" -databaseserver app1 -databasename Contoso\_WSS\_Content\_Portal\_01 -sitetemplate spsportal -description "Contoso Portal" -sethostheader -apidname PortalAppPool -apidtype configurableid -apidlogin contoso\portalapid -apidpwd P@ssw0rd

# Installing on Subsequent Servers in the Farm

Once you have installed on the first server in the farm, you need to *connect* from every subsequent server. To connect to an existing farm, you need to specify the SQL Server instance and configuration database, like this:

psconfig.exe -cmd configdb -connect -server app1 -database SharePoint\_Config\_Contoso -user contoso\mossfarm -password P@ssw0rd

You also need to install services, features, secure resources, and start services as you did before. The following example starts all services, features, and starts/configures the query role on the WFE Server(s):

psconfig.exe -cmd services install

psconfig.exe -cmd installfeatures

:: Setting Security on Registry and File System

psconfig.exe -cmd secureresources

stsadm.exe -o osearch -action start -role query -farmcontactemail administrator@contoso.msft -farmperformancelevel maximum -farmserviceaccount "contoso\mossservice" -farmservicepassword P@ssw0rd

stsadm.exe"-o osearch -action start -role query -farmcontactemail administrator@contoso.msft -farmperformancelevel maximum -farmserviceaccount "contoso\mossservice" -farmservicepassword P@ssw0rd -propagationlocation c:\indexes

# Summary

Always test farm installation scripts against a test server farm before using in production. Scripts rarely work correctly the first time – it is an iterative process. If using scripts for disaster recovery purposes, be sure they work the first time. Don’t depend on quesses as to where the script went wrong. Be aware some things are not backed up, such as Alternate Access Mapping, and must be manually added or scripted after a failure.

As always, use these examples at your own risk. Test, test, and then test some more! You will notice some scripts named *myhh.cmd, portalhh.cmd,* and *ssphh.cmd*. These scripts use *translate.js* to parse the IIS metabase and add host headers, if necessary. We wanted a *complete* farm install, including IIS settings. Changed the *<name>.cmd*  file to add host headers relevant to your server farm.