

Desired State Configuration

What is Desired State Configuration?

First off let's really "define" what DSC is. DSC is an extension of PowerShell, in that it contains all the needed Powershell cmdlets and logic needed to enforce configurations on systems in your environment. This is done via the creation of a MOF file ([Management Object File](#)), and that MOF file is either pushed to the target system, or pulled by the target system as we'll see in one of the upcoming posts.

Due to the fact that DSC is based off of PowerShell, those people who are quite familiar with PowerShell already, will find that configuring DSC is quite simple. This is not to say that those NOT familiar with PowerShell CAN'T do it. I'm simply saying it's easier if you already possess some sort of baseline PowerShell know-how.

Continuing on, the real goal of DSC is to enforce a particular "state" or configuration onto the target system, to ensure that it stays that way at all times. DSC is relatively new in the Microsoft stack, but it's become quite the topic of interest and is seen by many as the potential future of Windows Server setup and configuration.

With that in mind it isn't difficult to see that DSC's potential use-cases can be of real benefit to any Hyper-V administrator.

Benefits of Desired State Configuration

When it comes to Hyper-V, the host systems themselves are really the key players in our infrastructure. Sure, System Center VMM is the Quarterback in most situations, but the hosts are the workhorses. When running virtualized workloads, the hosts are of paramount importance, and as Hyper-V administrators it's our [jobs](#) to make sure they are configured and setup properly. DSC not only insures that the configuration is maintained and enforced once the host is established, but it can also make the roll-out of additional hosts into a cluster much easier.

When it comes to clustering, it's mostly true that all hosts in a given failover cluster are generally the same. DSC can help us during host deployments by forcing any new hyper-v server to adhere to the MOF file(s) we create. This will insure that the software configuration is the same on the new host as it is on the others in the cluster, and it really speeds up the process, because now you don't have to go and manually deploy all those settings. In addition, DSC can insure certain configurations are maintained on an ongoing basis.

An example I like to use is the case of a junior admin. Let's say you have a junior admin install AD or DNS on a Hyper-V host just because they felt like it. Not a good idea right? If you're using DSC it can and will come to your rescue if it's configured properly.

The next time DSC looks at the MOF file associated with that Hyper-V host its going to say “Whoa!... These roles aren’t supposed to be here... GET OUTTA HERE!” It then removes the offending roles so the host is in compliance with the defined MOF file. Thus making sure your hosts are just the way they’re supposed to be.

All-in-all, it’s not hard to see why DSC is becoming such a popular tool.

Requirements

So far, we’ve looked at what DSC is, and what its benefits are, but what do we need to use it in our environments? I’ve included a list of requirements below.

- **Windows [Management](#) Framework 4.0** – Needed due to the fact that this package contains PowerShell 4.0 and the associated cmdlets and providers.
- **PowerShell Remoting Enabled**
- **Windows Server 2008 R2 SP1 or Newer**
- **Windows 7 SP1 or Newer**
- **.Net 4.5**