

DESIRED STATE CONFIGURATION AND WHY WE CARE

SETTING CONTEXT AROUND DSC & GROUP POLICY



By Kevin Sullivan

[SDM Software, Inc.](#)

July, 2016

DESIRED STATE CONFIGURATION INTRODUCTION

Though Desired State Configuration (DSC) has been around for a while it was first introduced in PowerShell v4 with little fanfare. It is easy to imagine that there was quite a bit for the engineering team at Microsoft to figure out. How DSC was introduced to the world is an example of how Microsoft is changing dramatically from the company's roots. Most think these changes are for the better, such as how bits are delivered to the world, how the community is engaged, and what levels of participation are welcome. And many other pieces of the process are evolving, though not all for the best. The benefit of the earlier, traditional waterfall method of engineering was that one could expect core features to be articulated well and slightly documented prior to even making it to a beta group or a TAP organization. That is not the case with new PowerShell and DSC resources, so it's easy to feel a little lost. The positive side of this is that community participation is so much higher. This means if you are trying to figure out early features of something, you likely have others who are willing to jump in and help out.

This paper intends to define DSC, describe where it may impact IT, and show where it works with Group Policy and where it doesn't. While DSC uses a different technology approach to Group Policy, it is trying to accomplish some of the same goals--namely, manage Windows configuration.

DEFINITION

From the MSDN article ([DSC on MSDN](#)), "DSC is a new management platform in Windows PowerShell that enables deploying and managing configuration data for software services and managing the environment in which these services run." There is quite a bit of confusing stuff in that first statement, but the MSDN doc goes deep into describing scenarios and the individual pieces that make up DSC. You will want to take some time to explore.

One of the interesting things you'll find on the MSDN page is contributors:

Windows PowerShell Desired State Configuration Overview



The content around DSC and PowerShell is now welcoming members of the community to contribute. This is a discussion for another paper, but as Bob Dylan says, "The Times They Are a-Changin'." The MSDN page links directly to their github pages where you can find amazing stuff.

The page lists some of the common scenarios and applications for DSC. Take a look, but at the end of the day it's pretty simple: Define an administrator's or organization's intent (configuration) and deploy that intent to managed systems.

HELPFUL LINKS AND POINTERS

There's a great deal of information out there. Microsoft has provided a lot, and its small but active community has contributed a huge amount.

After wrapping your head around the key scenarios and getting a bit of a plan in your head you will likely want to dive deeper and start real learning. To me, there are two places to go for formal training/discussions on technologies like this: The Microsoft Virtual Academy and Pluralsight. Both have a lot to offer and are worth checking out. Lastly, as mentioned above, find folks who are in deep and explore their blogs and git pages. There is so much going on with DSC right now it can be a bit overwhelming. Take a deep breath, buckle up and enjoy the ride.

LINKS

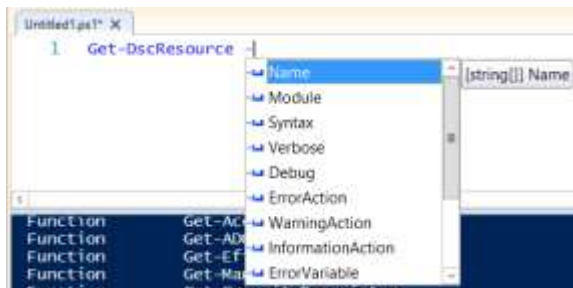
Check out these links for additional information (search for Desired State Configuration);

- Training - <https://mva.microsoft.com>
- Training - <https://www.pluralsight.com>
- Product Docs - <https://msdn.microsoft.com/en-us/PowerShell/dsc/overview>
- Product Docs - <https://technet.microsoft.com>
- Community - <https://github.com>
- Community - <http://www.powershellgallery.com>

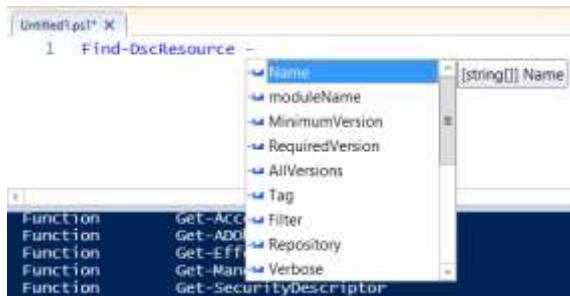
DSC SPECIFIC RESOURCES

Much of the magic of DSC is driven through "Resources." These DSC Resources provide the instructions that communicate with the target system on how to configure a specific resource, such as log files, registry values, files, services, processes, and more. Microsoft provides many resources out of the box for core scenarios, and the community is constantly publishing more to git or to the PowerShell Gallery. There are community resources, Microsoft resources, and experimental resources. . Experimental resources are those provided by Microsoft but not officially released for production use.

If you want to locate all DSC resources present on your local system, use `Get-DSCResource`;



If you want to locate all DSC resources that are present on registered repositories (like PowerShell Gallery), use `Find-DSCResource`;



You'll find a lot. At the time of this writing the PowerShell Gallery returned more than 650 DSC Resources. It will take some time to organize and determine which resources are important to you and the organization you are managing. There is quite a bit of talk about creating one's own DSC Resource Repository ([Private Gallery](#)). This concept is worth [checking out](#), particularly if you are in a regulated and process driven organization. Spend some time working with these cmdlets. Being able to verify version numbers is worth the price of admission alone!

While you're at it, take some time to peruse the PowerShell Gallery and GitHub repositories. Many community members are writing custom resources and sharing them broadly. GitHub is where a lot of the magic (i.e., work) is happening. When the resources are ready for prime time you may find them published to the PowerShell Gallery.

COMMON DSC SCENARIOS

Some of you may have been following DSC development for a long time. It has evolved quite a bit since it first appeared. In the early days the examples used to describe how DSC works were focused on deploying a web server. While this example seems simplistic now, given the broad abilities of DSC for Windows configuration, it served a useful purpose. The example of a web server is quite interesting because it covers multiple elements that need to be considered.

Deploying and maintaining a specific role based server, such as a web server, is possibly the #1 scenario for DSC. You can use DSC to declare the desired configuration for a given server role. That declaration is signed off by stake holders and confirmed as the 'de jure' policy. DSC provides the ability to ensure that policy is in place by not only deploying the settings but by regularly checking settings, remediating errant changes, and keeping the universe in balance. It is a wonderful thing.

Instead of posting some actual DSC configuration it can be more effective to show how to find the syntax of the configuration for a given DSC resource. This paper is designed to whet the appetite, not bog you down by trying to teach how to do everything. There is a great PowerShell parameter available when you are looking for DSC resources called '-Syntax'. It essentially shows you exactly how to author the given resource.

```
Get-DscResource -Name registry -Syntax
```

```
Registry [String] #ResourceName
{
    Key = [string]
    ValueName = [string]
    DependsOn = [string[]]
    Ensure = [string]{ Absent | Present }
    Force = [bool]
    Hex = [bool]
    PSDscRunAsCredential = [PSCredential]
    ValueData = [string[]]
    ValueType = [string]{ Binary | Dword | ExpandString | MultiString | Qword |
String }
}
```

Now you can simply manipulate the resource definition to meet your needs. In the absence of higher fidelity tooling, this is not bad. Takes some getting used to, but persist. Your invested time will return some valuable skills.

WHY DO WE CARE?

DSC is the configuration platform for today, for the cloud, for heterogeneous environments -- and it is here to stay. It has been in development for the better part of 10 years and is ready to get out there and have the tires kicked. Many organizations are already using DSC in their production environments. DSC will be a part of the configuration management environment for all organizations invested in Windows Technologies, including Azure. It will cover many configuration management (CM) scenarios. And you can be sure that there is a path from current CM technologies like Group Policy, to adoption of DSC, for many organizations. It's a path that IT architects are thinking about, and it's a path that SDM Software intends to help you follow as your own organization adopts DSC.

Today the vast majority of organizations invested in Microsoft technologies use Group Policy. Many organizations also use System Center for Desktop management and app deployment, including Windows Intune (Enterprise Mobility). IT professionals have worn multiple hats and continue to cross the line between Desktop/Device management and Server management.

Regardless of your platform and your target systems there are common technology needs. Reporting, Auditing, Compliance and Automation are consistently the most important. These four areas have been our focus at SDM Software for years. These needs will not change when the underlying platform for CM changes or evolves. Also, regarding your target, some core scenarios cross over between Desktop and Server: Service Management, Registry configuration, File deployments, script execution and much more. In addition, user centric configuration will continue to be an important challenge for folks. Role Based Config makes good business sense.

DSC was developed as a technology to address Server configuration needs. Although there are signs that Microsoft is loosening their story here, it is still primarily focused on configuring servers (Azure hosted, on-prem, Nano etc.). This server-specific emphasis is rooted in the adoption of DevOps patterns for IT applications management that has become increasingly popular, and in the application deployment model of Continuous Integration/Continuous Deployment (CI/CD) that is helping organizations efficiently roll out new and updated server applications. These new ways of deploying

server applications require a CM framework that meets the need for repeatable, deterministic, end-to-end automation—a framework like DSC.

SDM Software cares about this for the same reason that we care about Group Policy. As new patterns emerge to configure and deploy Windows, we want to be there, to provide solutions for Reporting, Auditing, Compliance and Automation that follow these Operational Management and DevOps-style workflows. Our focus on these core areas will continue, and we will incorporate new areas of attention to assist organizations during their co-existence and migration phases. We understand the need to balance core technologies with high fidelity tools to help IT professionals regardless of their skill development. As an example, a business goal might be a detailed report analyzing some configuration data, which may require an intuitive UI that makes that process simple. Some team members may be PowerShell gurus and want to script the solution. Achieving your business goal is the requirement -- how you do it should be up to you and not limited by the solution/platform you choose.

SUMMARY

Desired State Configuration is here now and it is time to learn what it means to you and your organization. Group Policy is a prevalent reality and it must be managed and maintained in an efficient way. Both approaches to Windows Configuration Management often require knowledge of techniques and solutions to deliver the full benefit of these approaches. SDM Software is an expert in configuration management, and will continue to innovate in this evolution of Configuration Management to help you succeed in your business goals around IT. We will also expose new and interesting scenarios to help teams simply do better--leveraging technologies like Group Policy and DSC with our many years of combined experience to help you get the most of your Windows infrastructure. This paper gives a brief introduction to Desired State Configuration, in relation to Group Policy.

SDM Software provides solutions around Reporting, Auditing, Compliance and Automation for your configuration environment. To get a glimpse of how we are beginning to help you make the transition to DSC, please take a look at our DSC Group Policy Client Side Extension on the freeware page of our website (<https://www.sdmssoftware.com/gpoguy/free-tools/library/>) and our PowerShell function to migration registry based Group Policy settings to DSC resources (available at www.github.com/gpoguy).