<https://www.petri.com/how-to-use-the-dism-tool-to-manage-windows-features?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A%20Petri%20%28Petri%20IT%20Knowledgebase%29>

**[How to use the DISM Tool to Manage Windows Features](http://feedproxy.google.com/~r/Petri/~3/zhXdCBFP260/how-to-use-the-dism-tool-to-manage-windows-features?utm_source=feedburner&utm_medium=email)**



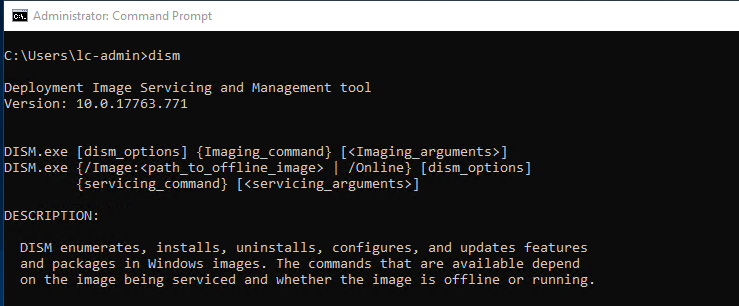
The [**Deployment Image Servicing and Management (DISM)**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/dism---deployment-image-servicing-and-management-technical-reference-for-windows) command-line tool and associated Powershell module are powerful tools to manipulate Windows image .wim files, virtual hard disks .vhd files, and control of Windows features and drivers.

Even more powerful is the ability to service offline images. In the past, there were many utilities that were used, but DISM is intended to replace several such as PEimg, Intlcfg, ImageX, and Package Manager.

## Using DISM

To start with, open an administrative command prompt and simply type dism. Initial output is the version and options available, of which there are many.

Image #1 Expand



In this article, we are focused on how to use DISM to specifically add or remove Windows features. With that in mind, let’s explore what functionality exists to enable this.

## DISM Feature Functionality

There are four main functions that are used with managing features. Below we will run through each one and explore exactly how they are used.

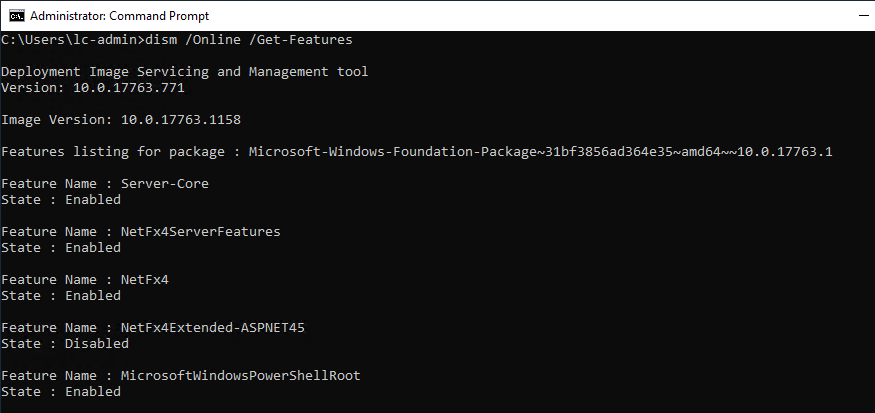
* [**Get-Features**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/dism-operating-system-package-servicing-command-line-options#get-features)
* [**Get-FeatureInfo**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/dism-operating-system-package-servicing-command-line-options#get-featureinfo)
* [**Enable-Feature**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/dism-operating-system-package-servicing-command-line-options#enable-feature)
* [**Disable-Feature**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/dism-operating-system-package-servicing-command-line-options#disable-feature)

### Get-Features

To understand what features are available, and the current state, we use the /Get-Features parameter. It’s important to note that we are using /Online for all of the commands here as we are operating on the current operating system and not an offline image.

dism /Online /Get-Features

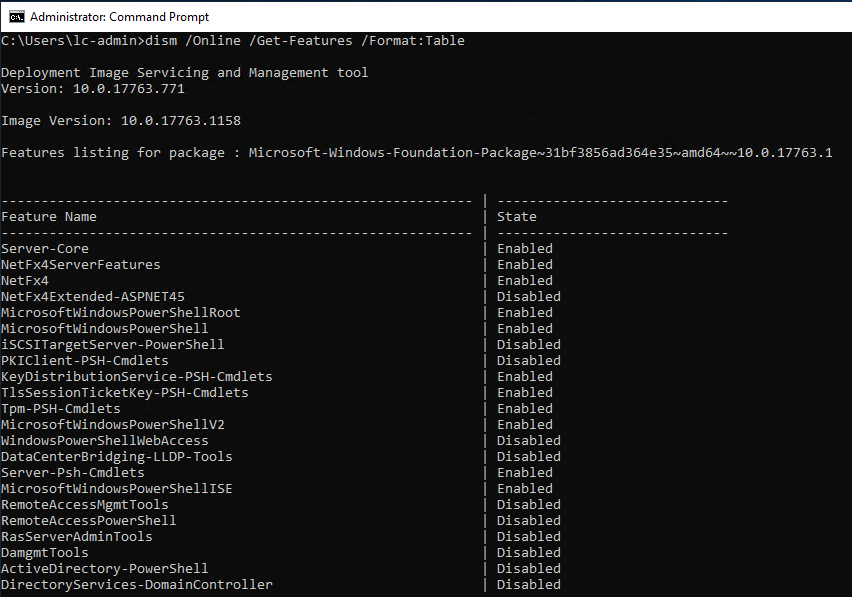
Image #2 Expand



When this command is run, an output of all features and their current state is shown on screen. There are many and the default List view can be difficult to read. Thankfully there is a convenient format option that makes the output easier to parse.

dism /Online /Get-Features /Format:Table

Image #3 Expand



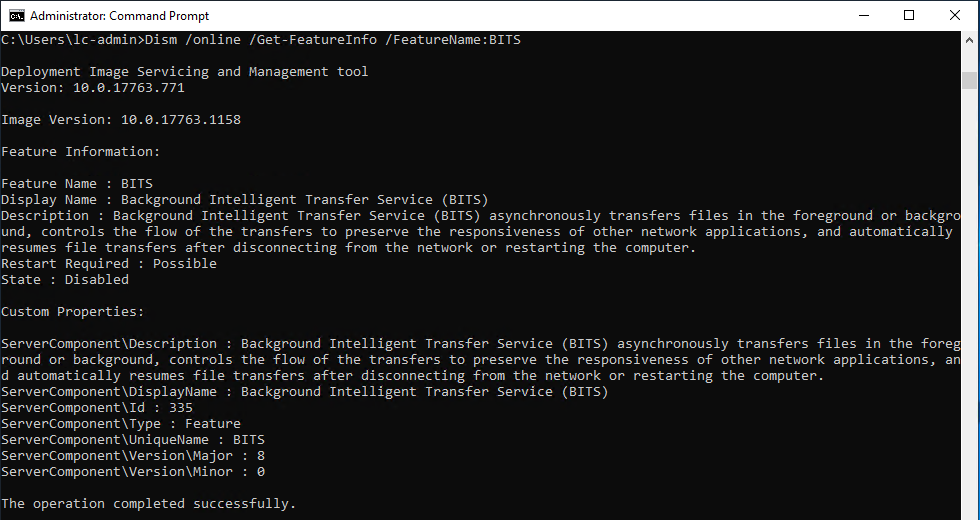
Now that we know what features are available, let’s pick one in particular to enable.

### Get-FeatureInfo

To learn more about a specific feature, we need to use the Get-FeatureInfo option. This command requires an additional parameter as well, FeatureName that contains the name of the feature to query. In this case, we are looking for the BITS (Background Intelligent Transfer Service) feature.

dism /online /Get-FeatureInfo /FeatureName:BITS

Image #4 Expand



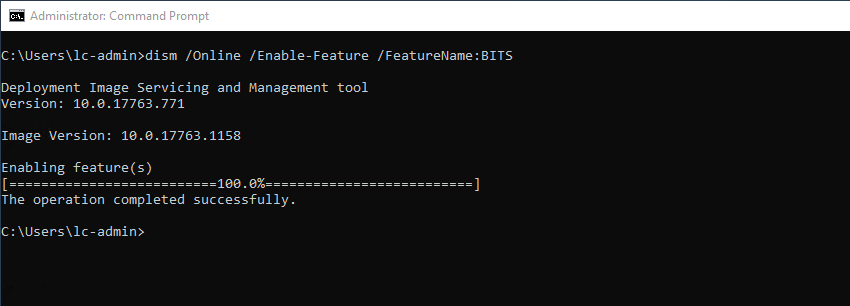
There are a couple of useful results returned. Notably, the Restart Required and Version properties. This is useful to know how installing the feature will affect the system and if you have the correct version. Next is enabling the feature on the system, to do that we use the Enable-Feature option.

### Enable-Feature

If the feature that you want to enable has the sources already included in the operating system and is a stand-alone feature, such as BITS, then the command is very simple.

dism /Online /Enable-Feature /FeatureName:BITS

Image #5 Expand



There are a few other commands that are very useful.

* **Source** – /Source:X:\\WindowsSources\\SxSIf the sources are not available for the requested feature, you can specify a path. This is common for features such as .NET 3.5.
* **LimitAccess** – /LimitAccessIf you want to prevent DISM from querying Microsoft Update for sources, this will prevent DISM from doing just that.
* **All** – /AllWithout specifying /All, only the requested feature will be installed, but using All will allow all parent features to be installed as well. This can help with dependency management sometimes.

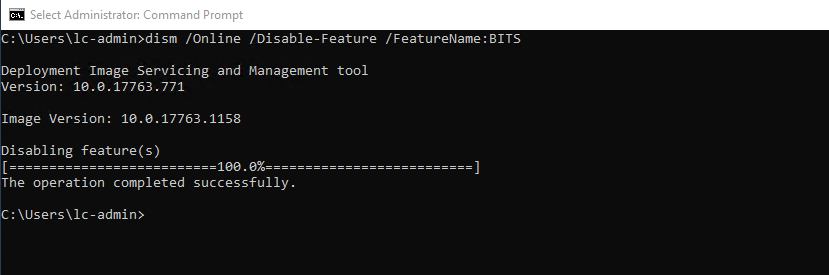
Finally, perhaps it’s been decided that using BITS is not necessary, therefore we need to remove the feature from the operating system.

### Disable-Feature

To simply remove a feature, it’s essentially the same as Enable-Feature but the inverse.

dism /Online /Disable-Feature /FeatureName:BITS

Image #6 Expand



There is one additional option that can be used which is /Remove. This will remove the feature but not the features manifest. This is typically used to free up space in an image. Upon installation request, a remote source would need to be specified to install the feature.

### Petri NewslettersOffice 365 Insider

Our Petri Office 365 Insider is dedicated to sharing detailed knowledge from top Office 365 experts. Delivered once a month to your inbox.

[Submit]

[**All Newsletters**](file:///\\fs\manage-subscriptions)

Petri.com may use your contact information to provide updates, offers and resources that may be of interest to you. You can unsubscribe at any time. To learn more about how we manage your data, you can read our [**Privacy Policy**](https://www.bwwmediagroup.com/terms-and-conditions/) and [**Terms of Service**](https://www.bwwmediagroup.com/terms-and-conditions/).

!Already a Petri.com member? [**Login here**](file:///\\fs\login) for 1-click registration.

## DISM Powershell Module

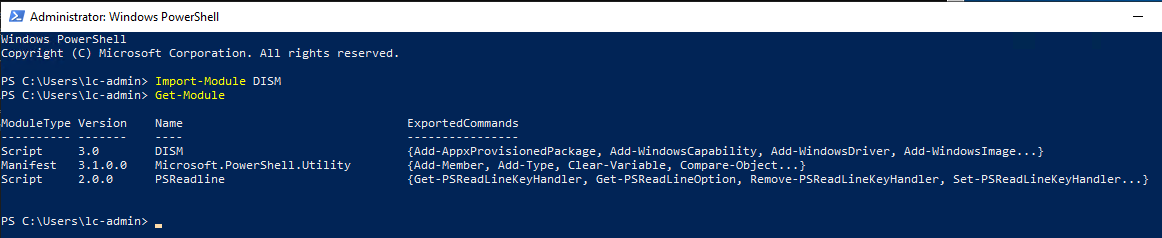
Using PowerShell, you can access many of the [**same features of DISM**](https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/use-dism-in-windows-powershell-s14) but in a PowerShell, object-centric, manor. Specifically, the features map as below.

* /Get-Features → Get-WindowsOptionalFeature
* /Get-FeatureInfo → Get-WindowsOptionalFeature -FeatureName
* /Enable-Feature → Enable-WindowsOptionalFeature
* /Disable-Feature → Disable-WindowsOptionalFeature

Let’s quickly walk through how these features work in comparison to the standard command-line DISM tool.

You may have to import the DISM module, Import-Module DISM

Image #7 Expand



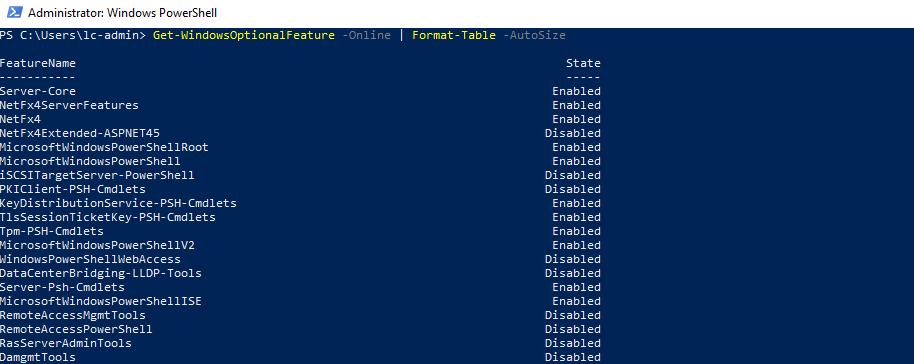
Once the module is loaded, there are many commands but we will focus on the four covered in this article.

### Get-WindowsOptionalFeature

One immediate difference is that there is no Format parameter. This is because you are able to format the returned objects in any number of ways using PowerShell. The way to get a similar table view is to do the following.

Get-WindowsOptionalFeature -Online | Format-Table -AutoSize

Image #8 Expand



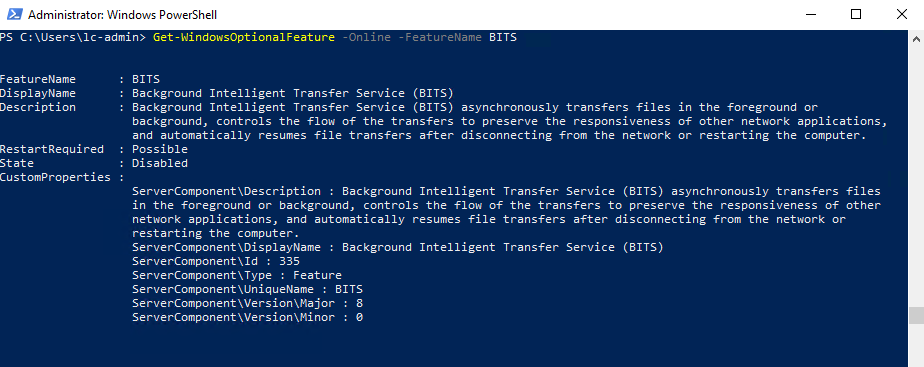
Next, we will want to learn more about an existing feature.

### Get-WindowsOptionalFeature -FeatureName

You will notice that the cmdlet used is the same as before, but using a parameter instead. Due to how PowerShell is generally structured, this is the preferred method. To do this we will pass in the feature that we want to learn more about.

Get-WindowsOptionalFeature -Online -FeatureName BITS

Image #9 Expand



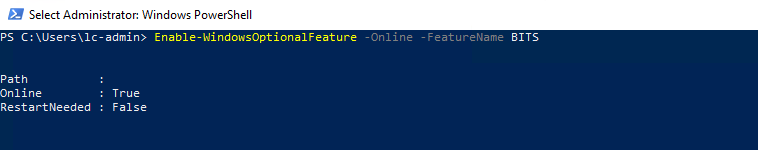
As you can see, the same information that was returned by DISM is contained in an object returned in PowerShell. Now that we have our feature to add, let’s Enable this feature.

### Enable-WindowsOptionalFeature

There are several ways to use this command. The most analogous to the DISM command-line tool is the following.

Enable-WindowsOptionalFeature -Online -FeatureName BITS

Image #10 Expand



If you have a series of features or know the one you want, you can use the pipeline to pass these in. Below are two examples of how to do this.

# Pass an array of features to be enabled

@("BITS") | ForEach-Object { Enable-WindowsOptionalFeature -Online -FeatureName $\_ }

# Retrieve the feature first and then Enable the feature

Get-WindowsOptionalFeature -Online -FeatureName BITS | Enable-WindowsOptionalFeature -Online

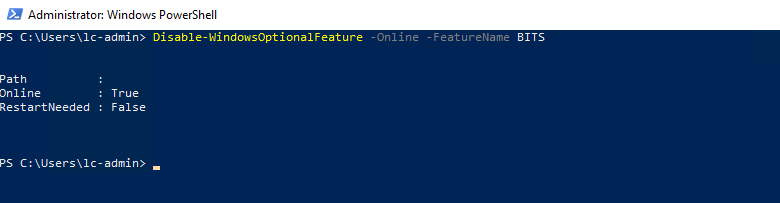
Finally, we need to disable BITS as it is no longer needed.

### Disable-WindowsOptionalFeature

Just like the enable cmdlet, we simply need to pass the feature name to disable.

Disable-WindowsOptionalFeature -Online -FeatureName BITS

Image #11 Expand



## Conclusion

Whether you use DISM through the command-line or through PowerShell, there is a lot of functionality available to manage Windows Features. Not only can DISM be used on a running system, but if you have offline images, it can be used to manage those as well!