

Readme

WASM Cmdlets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Version: | 1.0.0 | | Last updated: | 4/19/2011 | |  |

## Contents

[Overview 2](#_Toc290985910)

[Get Started 2](#_Toc290985911)

[Task 1 - Build and Install the Windows Azure Service Management Tools 2](#_Toc290985912)

[Task 2 - Listing Installed Cmdlets 5](#_Toc290985913)

[Task 3 - Displaying CmdLets Help 6](#_Toc290985914)

[Using the Cmdlets 8](#_Toc290985915)

[Task 1 - Using the Windows Azure Hosted services cmdlets 10](#_Toc290985916)

Overview

* 1. The Windows Azure Service Management Tools includes a set of cmdlets that enable a user to configure and manage several Windows Azure operations including: Azure Services and Azure Storage.
  2. These tools can be helpful when developing and testing applications that use Windows Azure Services. For instance, using these tools you can easily create a new deploy of your services, change configuration for a specified role, etc.

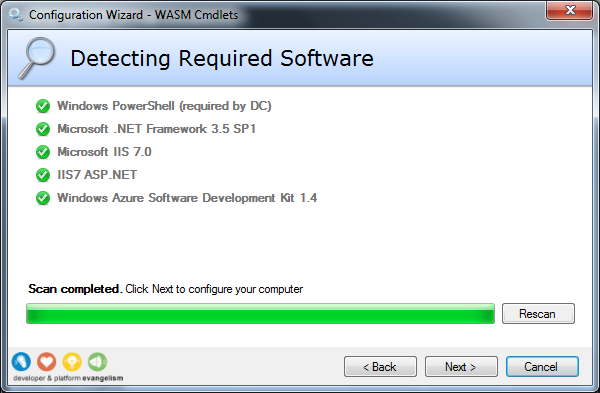
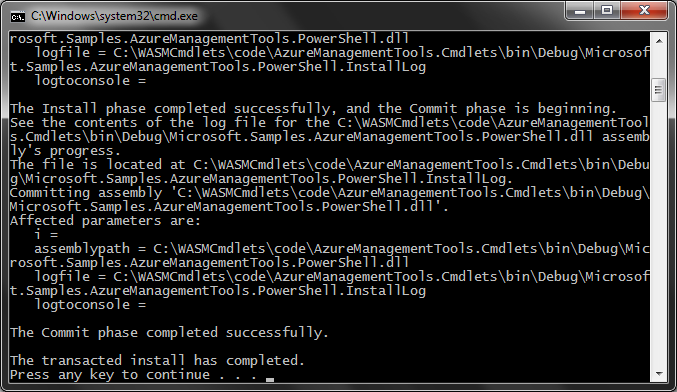
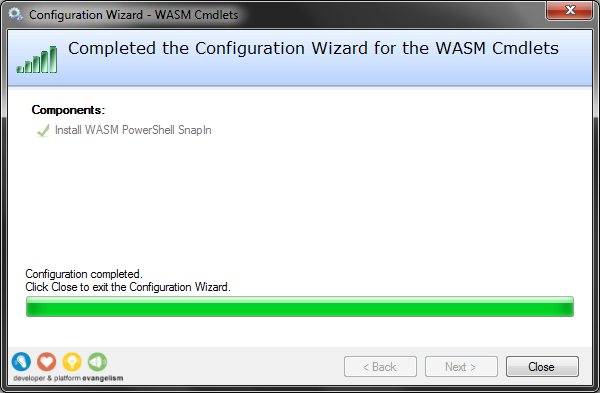
# Key Features

* 1. The Windows Azure Service Management Tools are designed to be usable tools for browsing, configuring, and managing several of the Windows Azure Services. However, they are also provided in source code form to enable you to better understand how to use the Windows Azure Services Platform. Some of the key features demonstrated include:
  + Windows Azure Service Management API
  + Windows Azure SDK
  + Windows Powershell

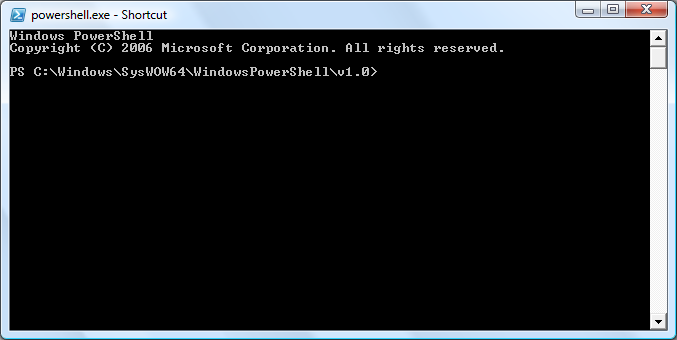
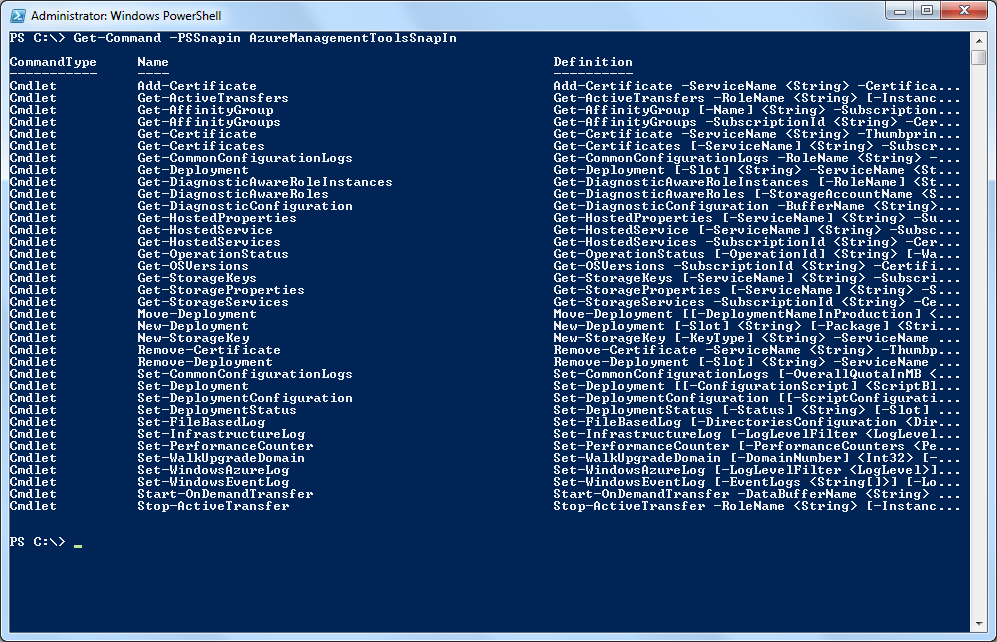
Get Started

* 1. The Windows Azure Service Management Tools are provided in source code form. Before you can use the tools they need to be compiled and installed.

Task 1 - Build and Install the Windows Azure Service Management Tools

* 1. To get started using the Azure Services Management, complete the following steps:
  2. Run the **StartHere.cmd** command script located in the directory where you extracted the Windows Azure Service Management Tools package.
  3. The **StartHere.cmd** script will launch the Configuration Wizard. The Configuration Wizard is designed to check your machine to ensure that it is properly configured with all of the dependencies to build and use the Azure Services Platform Management Tools.
     1. 
     2. Figure 1
     3. Configuration Wizard
  4. The next step involves checking your machine for the required software and configuration. If you don’t have the require configuration or dependencies, then in most cases you will be provided with a link to download them.
     1. 
     2. Figure 2
     3. Checking Dependencies
  5. After the dependency check is complete, press the **Next** button to proceed in the Configuration Wizard.
  6. The next step in the Configuration Wizard involves building and installing the Windows Azure Service Management Tools. The Configuration Wizard will execute the **installPSSnapIn.cmd** command script located in the **\setup\scripts\tasks** folder where you extracted the Windows Azure Service Management Tools.
     1. **Note:** The \setup\scripts\tasks folder for this tool includes scripts for building, installing, and uninstalling the Azure Services Management Powershell SnapIn. You can run these files in case you need to troubleshoot a problem with the automated Configuration Wizard.
     2. 
     3. Figure 3
     4. Build Task
  7. Finally, after building and installing the Windows Azure Service Management Tools, the Configuration Wizard will be complete.
     1. 
     2. Figure 4
     3. Configuration Wizard Completed

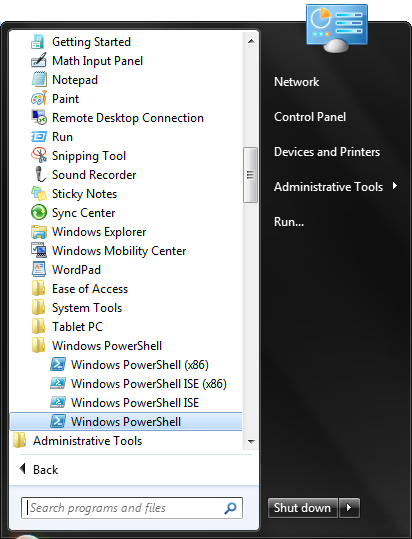
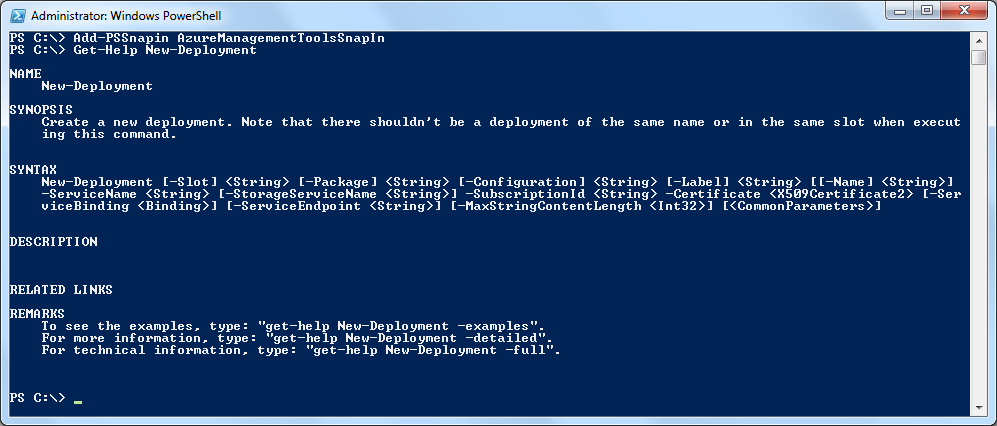
Task 2 - Listing Installed Cmdlets

* 1. The Azure Management Tools is a set of Windows PowerShell cmdlets for performing all of the operations related with administration and management of Azure services and storage.
  2. The set of PowerShell cmdlets provided with ECF can be retrieved using a Powershell script:
  3. Open PowerShell console from **Start | All Programs | Accessories | Windows PowerShell | Windows PowerShell**. The PowerShell console should open.
     1. 
     2. Figure 5
     3. PowerShell Console
  4. Add the AzureManagement snapin typing the following command:
     1. PowerShell
     2. Add-PSSnapin AzureManagementToolsSnapIn
  5. Retrieve all cmdlets typing the following command:
     1. PowerShell
     2. Get-Command -PSSnapin AzureManagementToolsSnapIn
     3. 
     4. Figure 6
     5. Listing the available cmdlets

Task 3 - Displaying CmdLets Help

When using individual cmdlets you will probably need to know detailed information about how to use each cmdlet.

To get help information about a particular cmdlet follow the next steps:

* 1. Open PowerShell console from **Start | All Programs | Accessories | Windows PowerShell | Windows PowerShell**. The PowerShell console should open.
     1. 
     2. Figure 7
     3. Opening the PowerShell console
  2. Add the **AzureManagementTools** snapin typing the following command:
     1. PowerShell
     2. Add-PSSnapin AzureManagementToolsSnapIn
  3. Retrieve help information using the **Get-Help** command. For example, to get help for the **New-Deployment** cmdlet you can type:
     1. PowerShell
     2. Get-Help New-Deployment
     3. 
     4. Figure 8
     5. Retrieving help information for the New-Deployment
  4. **Note:** To get more detailed information about a particular cmdlet you can execute the **Get-Help** cmdlet with the **–Detailed** or the **–Full** option. This will give information about parameters, usage examples, etc.
  5. For example:
  6. Get-Help Move-Deployment –Detailed

Using the Cmdlets

* 1. The Windows Azure Service Management Tools PowerShell cmdlets can be used to run unattended scripts to configure and manage the Azure Services. The PowerShell cmdlets provided with this package include the following:
  2. **Windows Azure Hosted Services**

|  |  |
| --- | --- |
| Name | Description |
| Get-Deployment | View details of a specified deployment. |
| Get-HostedProperties | List the properties for the specified hosted account. |
| Get-HostedService | Retrieve a specified hosted account. |
| Get-HostedServices | Lists all hosted services underneath the subscription. |
| Move-Deployment | Swaps the deployments in production and stage. |
| New-Deployment | Create a new deployment. Note that there shouldn't be a deployment of the same name or in the same slot when executing this command. |
| Remove-Deployment | Deletes the specified deployment. Note that the deployment should be in suspended state. |
| Set-Deployment | Initiates an in-place upgrade of the specified deployment. |
| Set-DeploymentConfiguration | Change the deployment's configuration. |
| Set-DeploymentStatus | Change deployment status to running or suspended. |
| Set-WalkUpgradeDomain | Walks the specified upgrade domain. |
| Get-OSVersions | Lists the versions of the guest operating system that are currently available in Windows Azure. |

* 1. **Windows Azure Storage**

|  |  |
| --- | --- |
| Name | Description |
| Get-StorageKeys | Displays the primary and secondary keys for the account. Should have  the storage account resource specified. |
| Get-StorageProperties | List the properties for the specified storage account. |
| Get-StorageServices | Lists all storage services underneath the subscription. |
| New-StorageKey | Regenerates storage keys with the key-type parameter specifying which key to regenerate. Should have the storage account resource specified. |

* 1. **Windows Azure Affinity Groups**

|  |  |
| --- | --- |
| Name | Description |
| Get-AffinityGroup | List the properties for the specified affinity group. |
| Get-AffinityGroups | Lists all affinity groups in the subscription. |

* 1. **Windows Azure Service Certificates**

|  |  |
| --- | --- |
| Name | Description |
| Get-Certificates | List the certificates for the specified hosted service. |
| Get-Certificate | Retrieve a specified service certificate. |
| Add-Certificate | Upload a service certificate. |
| Remove-Certificate | Deletes the specified service certificate. |

* 1. **Windows Azure Diagnostics**

|  |  |
| --- | --- |
| Name | Description |
| Get-ActiveTransfers | Returns the set of active transfers, with associated transfer information. |
| Get-CommonConfigurationLogs | Gets the common configuration values for all logging buffers. |
| Get-DiagnosticAwareRoleInstances | Returns a list of IDs of active role instances that have a diagnostic monitor running. |
| Get-DiagnosticAwareRoles | Lists the set of roles which have successfully started at least one diagnostic monitor. |
| Get-DiagnosticConfiguration | Gets the buffer configuration for the specified buffer name. |
| Set-CommonConfigurationLogs | Sets the common configuration values for all logging buffers. |
| Set-FileBasedLog | Sets the buffer configuration for file-based logs. |
| Set-InfrastructureLog | Sets the buffer configuration for the logs generated by the underlying diagnostics infrastructure. The diagnostic infrastructure logs are useful for troubleshooting the diagnostics system itself. |
| Set-PerformanceCounter | Sets the buffer configuration for performance counter data. |
| Set-WindowsAzureLog | Sets the buffer configuration for basic Windows Azure logs. |
| Set-WindowsEventLog | Sets the buffer configuration for Windows event logs. |
| Start-OnDemandTransfer | Starts an on-demand transfer of the specified data buffer. |
| Stop-ActiveTransfer | Stops active on-demand transfer of the specified transfer id. |

Task 1 - Using the Windows Azure Hosted services cmdlets

* 1. In this task, you will see a few examples demonstrating how to use the PowerShell cmdlets for accessing and managing Azure Hosted services.
  2. The first step is to get hold of a valid X509 certificate with a key size of at least 2048 bits. One quick way is to use **makecert.exe** (which ships with the Windows SDK) and use a command like the below:
     1. Command Line
     2. makecert -r -pe -a sha1 -n "CN=Windows Azure Authentication Certificate" -ss My -len 2048 -sp "Microsoft Enhanced RSA and AES Cryptographic Provider" -sy 24 testcert.cer
     3. **Note:** This command must be run under elevated privileges.
  3. Next, upload the **testcert.cer** file to the developer portal to let Windows Azure know that is should trust the certificiate for API operations on your project.
  4. Start Powershell if it is not already running by selecting **Windows Powershell** from the start menu.
  5. Within the Windows PowerShell command prompt, enter the following command to add the Azure Services Management cmdlets to the console’s scope, if you have not already done so:
     1. PowerShell
     2. Add-PSSnapin AzureManagementToolsSnapIn
  6. Now, let’s see how to deploy a new package using the PowerShell cmdlet. The following command demonstrates how to create a new deploy named TestDeploy on staging. You can get the subscription ID and API certificate thumbprint from Windows Azure portal.
     1. PowerShell
     2. New-Deployment -subscriptionId %SubscriptionId% -certificate (get-item cert:\CurrentUser\MY\%thumbprintInUpperCase%) -serviceName %serviceName% -slot staging -package http://%storageServiceName%.blob.core.windows.net/%container%/testPackage.cspkg -configuration config\TestServiceConfiguration.cscfg -name TestDeploy -label TestLabelStaging -storageservicename %StorageServiceName%
     3. **Note:** If the storage account name is the same as the hosted service name, the storageservicename parameter can be omitted.
  7. Run the following command to retrieve the deployment created in the previous step.
     1. PowerShell
     2. Get-HostedServices -subscriptionId %SubscriptionId% -certificate (get-item cert:\CurrentUser\MY\%thumbprintInUpperCase%) | where {$\_.ServiceName -eq "%serviceName%"} | Get-Deployment staging
  8. Finally, to delete a deploy you can use the Remove-Deployment cmdlet. The following example removes the deployment on staging. Deployment that we deployed in the previous steps:
     1. PowerShell
     2. Get-HostedService "%serviceName%" -subscriptionId %SubscriptionId% -certificate (get-item cert:\CurrentUser\MY\%thumbprintInUpperCase%) | Get-Deployment staging | Remove-Deployment