# Cmdlet Desginer Deployment Guide

## Prerequisites

1. Install Visual Studio 2008
2. Enable the IIS windows feature, plus “ASP.Net.” Under “Security,” enable Windows Authentication feature.
3. Install WPF toolkit: <http://www.codeplex.com/wpf/Release/ProjectReleases.aspx?ReleaseId=15598>
4. Compile the solution

## Create the Database

1. Open AspenManagementDB\createdb.sql
2. Search for FIXME\_PASSWORD, and select a strong database password. Replace FIXME\_PASSWORD with PASSWORD.
3. For a SQL Express installation, run:
   1. sqlcmd -S .\sqlexpress -i .\createdb.sql
   2. sqlcmd -S .\sqlexpress -i .\createverbs.sql
4. Enable SQL Authentication (for SQL Express, if needed)
   1. Set-ItemProperty 'HKLM:\software\microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer' LoginMode 2
   2. Restart-Service 'MSSQL$SQLEXPRESS'
5. Test a query:
   1. Invoke-SqlCommand .\SQLEXPRESS AspenManagement "SELECT \* FROM Verb" -Credential (Get-Credential)
      1. Use the AspenAdmin as the username
      2. Your password as the password
6. To refresh the database:
   1. Restart-Service 'MSSQL$SQLEXPRESS'
   2. Remove-Item "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\AspenAdmin\*"
   3. sqlcmd -S .\sqlexpress -i .\createdb.sql
   4. sqlcmd -S .\sqlexpress -i .\createverbs.sql
   5. Invoke-SqlCommand .\SQLEXPRESS AspenManagement "SELECT \* FROM Verb" -Credential (Get-Credential)
      1. This step is usually required, as SQL the connection is broken by the refresh. Connecting again gives you an error, and then the times after that return data.

## Deploy the Web service

1. Create a directory to contain the Cmdlet Designer-related sites
2. Copy the ‘AspenManagement’ folder to that directory
3. Update its web.config
   1. Update the connection string to point at the server hosting the database
   2. Update the password to match the password used in the database creation
4. Create the application in IIS
   1. Inetmgr.exe
   2. Right-click the default web site, “Add Application” named “CmdletDesigner”
   3. Disable anonymous authentication
   4. Enable Windows authentication
5. Test a webservice call:
   1. <http://localhost/cmdletdesigner/cmdlet.asmx?op=GetVerb>
   2. Use ‘%’ as the name, then press ‘Invoke’

## Install the Cmdlet Snapin

1. Go to AspenManagementCmdlets\AspenManagementCmdlets\bin\release
2. c:\windows\Microsoft.NET\Framework\v2.0.50727\InstallUtil.exe .\PsSpecCmdlets.dll
3. Test a cmdlet call:
   1. Launch PowerShell
   2. Add-PsSnapin PsSpecCmdletManagement
   3. Get-PsSpecVerb -ServiceUri <http://localhost/cmdletdesigner/cmdlet.asmx>

## Run the Cmdlet / WebService / Database Unit Tests

1. From AspenManagementCmdlets\UnitTests, run Test-Files.ps1

## Test the Cmdlet Designer

1. Visit cmdletDesigner\cmdletDesigner\bin\Release
2. Edit CmdletDesigner.exe.config, and uncomment the ServiceUri appSetting
3. Launch CmdletDesigner.exe
4. Click “Manage Projects”
5. Create a sample project, noun, cmdlet, etc, then delete them again.
   1. Note: deleting a project is not supported through the UI. To do so, open the PowerShell Automation Console, and type:  
      $uri = "http://localhost/cmdletdesigner/cmdlet.asmx"

Get-PsSpecProject -ServiceUri $uri | Remove-PsSpecProject -ServiceUri $uri

1. You can also run “GenerateData.ps1” from the PowerShell Automation Console

## Deploy the Spec Viewer

1. Copy the ‘SpecViewer’ folder to a directory (for example, the directory containing the web service)
2. Update its web.config
   1. Update ServiceUri to point to the deployed web service location
   2. Update AllowedHosts to include the machine the app is deployed on
3. Create the application in IIS
   1. Inetmgr.exe
   2. Right-click the default web site, “Add Application” named “SpecViewer”
   3. Disable anonymous authentication
   4. Enable Windows authentication
4. Ensure the PSSpecCmdlet snapin is registered on that system
5. Set the system-wide execution policy to “RemoteSigned”
   1. Set-ExecutionPolicy RemoteSigned
6. Test a spec:
   1. <http://localhost/SpecViewer/Default.aspx?Project=MySampleProject&Cmdlet=Set-MyProcess&ServiceUri=http://localhost/cmdletdesigner/Cmdlet.asmx>

# Data Migration

If the database schema needs to change between updates, SQL’s standard bcp utility does not handle data migration with a changing schema. The AspenManagementDB directory includes scripts to support data migration from an old database to a new one, so that all new databases can be built directly from scripts (rather than in-place schema updates that risk diverging from the checked-in version.)

1. Create a new database using the new createdb.sql
2. Update Generate-DataTransferSql.ps1 to account for tables and columns that should be skipped during the migration, or mapped to new column names.
3. Run Invoke-DataMigration.ps1 from the live server to the test server.
4. Validate data consistency between the old database and the new one
5. Promote the new database as the primary database.