Deploying command prompt steps:

------------- option 1 – did not work

"C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\IDE\devenv.exe" "C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014\FogBugz\_2014.dwproj" /build development /out ssasbuild.log

"C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\IDE\devenv.exe" "C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014.sln" /build development /out ssasbuild.log

------------ option 2

(for some reason, the Visual Studio 12.0 would not build the project, so use the 14.0 one)

"C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe" "C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014.sln" /Build Development /Project FogBugz\_2014

"C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\ManagementStudio\Microsoft.AnalysisServices.Deployment.exe" "C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014\bin\FogBugz\_2014.asdatabase" /d /o:"C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014\bin\FogBugz\_2014\_Script.xmla"

STEPS:

PROJECTLOCATION=”C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014\bin”

1. "C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\devenv.exe" "C:\Cubes\FogBugz\FogBugz\FogBugz\_2014\FogBugz\_2014.sln" /Build Development /Project FogBugz\_2014
   1. This generates an FogBugz\_2014.asdatabase file
2. Manually execute Deployment Wizard to get initial settings. Only need to do one time.
   1. This generates a FogBugz\_2014\_Script.xmla file
3. "C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\ManagementStudio\Microsoft.AnalysisServices.Deployment.exe" "PROJECTLOCATION\FogBugz\_2014.asdatabase" /d /o:"PROJECTLOCATION\FogBugz\_2014\_Script.xmla"
   1. This scripts the generation of the FogBugz\_2014\_Script.xmla file
4. "C:\Users\Longphi Nguyen\Downloads\temp\ASCMD\_StressTestingScripts\ASCMD\_StressTestingScripts\Ascmd.exe" -i "PROJECTLOCATION\FogBugz\_2014\_Script.xmla" -S **<SERVER>**

The \*.xmla script can be run with C# and AMO.net, as an ascmd command-line utility. Download from <https://sqlsrvanalysissrvcs.codeplex.com/releases/view/22769> the ASCMD\_StressTestingScripts package. The package contains Ascmd.exe.. Also make an Ascmd.exe.config file located by Ascmd.exe that contains

<configuration>

<runtime>

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

<dependentAssembly>

<assemblyIdentity name="Microsoft.AnalysisServices" publicKeyToken="89845dcd8080cc91" />

<bindingRedirect oldVersion="10.0.0.0" newVersion="12.0.0.0" />

<publisherPolicy apply="no" />

</dependentAssembly>

</assemblyBinding>

</runtime>

</configuration>

For **ascmd**, need:

Analysis Services

An instance of Analysis Services must be installed and running, because the **ascmd** command-line utilityis used to connect to an instance of Analysis Services and execute MDX queries, XMLA scripts, and DMX statements.

SQL Server Management Studio and Business Intelligence Development Studio

These two work environments provide supporting infrastructure for you to complete any task related to Analysis Services. For any given task, you can approach implementation either through the user interface or programmatically.

Analysis Management Objects (AMO)

AMO is required to execute the **ascmd** command-line utility on a computer that does not have Analysis Services installed.

.NET Framework 2.0

The .NET Framework 2.0 is required for the **ascmd** command-line utility to run.

------------- how to use devenv.exe

Use:

devenv [solutionfile | projectfile | anyfile.ext] [switches]

The first argument for devenv is usually a solution file or project file.

You can also use any other file as the first argument if you want to have the

file open automatically in an editor. When you enter a project file, the IDE

looks for an .sln file with the same base name as the project file in the

parent directory for the project file. If no such .sln file exists, then the

IDE looks for a single .sln file that references the project. If no such single

.sln file exists, then the IDE creates an unsaved solution with a default .sln

file name that has the same base name as the project file.

Command line builds:

devenv solutionfile.sln /build [ solutionconfig ] [ /project projectnameorfile [ /projectconfig name ] ]

Available command line switches:

/Build Builds the solution or project with the specified solution

configuration. For example "Debug". If multiple platforms

are possible, the configuration name must be enclosed in quotes

and contain platform name. For example: "Debug|Win32".

/Clean Deletes build outputs.

/Command Starts the IDE and executes the command.

/Deploy Builds and then deploys the specified build configuration.

/Edit Opens the specified files in a running instance of this

application. If there are no running instances, it will

start a new instance with a simplified window layout.

/LCID Sets the default language in the IDE for the UI.

/Log Logs IDE activity to the specified file for troubleshooting.

/NoVSIP Disables the VSIP developer's license key for VSIP testing.

/Out Appends the build log to a specified file.

/Project Specifies the project to build, clean, or deploy.

Must be used with /Build, /Rebuild, /Clean, or /Deploy.

/ProjectConfig Overrides the project configuration specified in the solution

configuration. For example "Debug". If multiple platforms are

possible, the configuration name must be enclosed in quotes

and contain platform name. For example: "Debug|Win32".

Must be used with /Project.

/Rebuild Cleans and then builds the solution or project with the

specified configuration.

/ResetAddin Removes commands and command UI associated with the specified Add-in.

/ResetSettings Restores the IDE's default settings, optionally resets to

the specified VSSettings file.

/ResetSkipPkgs Clears all SkipLoading tags added to VSPackages.

/Run Compiles and runs the specified solution.

/RunExit Compiles and runs the specified solution then closes the IDE.

/SafeMode Launches the IDE in safe mode loading minimal windows.

/Upgrade Upgrades the project or the solution and all projects in it.

A backup of these files will be created as appropriate. Please

see Help on 'Visual Studio Conversion Wizard' for more

information on the backup process.

Product-specific switches:

/componentsDiff Compares two component folders. Requires three arguments:

OptionalTfsServerCollectionUri SourceVersionedServerFolderPath

TargetVersionedServerFolderPath

If OptionalTfsServerCollectionUri is not provided, then an

attempt is made to use the current TFS project collection.

/debugexe Open the specified executable to be debugged. The remainder of

the command line is passed to this executable as its arguments.

/diff Compares two files. Takes four parameters:

SourceFile, TargetFile, SourceDisplayName(optional),

TargetDisplayName(optional)

/openProjects /scc:git Opens committed project from git.

Arguments:

/repositoryUrl (/repo):<RequiredRepositoryUrl>

/commitId (/id):<RequiredCommitId>

/project:<RequiredProjectItemPath>

/solution (/sln):<OptionalSolutionItemPath>

/displayId:<OptionalCommitDisplayId>

/timestamp:<OptionalCommitTimestamp>

/comment:<OptionalCommitComment>

/author:<OptionalCommitAuthor>

/openProjects /scc:tfs Opens versioned project from tfs.

Arguments:

/collection (/col):<OptionalTfsServerCollectionUri>

/version:<RequiredTfsVersionSpec>

/project:<RequiredProjectServerItemPath>

/solution (/sln):<OptionalSolutionServerItemPath>

If OptionalTfsServerCollectionUri is not provided, then an

attempt is made to use the current TFS project collection.

/sqldbaction Start SQL Server Data Tools and perform the action specified in the argument string.

/TfsLink Opens Team Explorer and launches a viewer for the

provided artifact URI if one is registered.

/useenv Use PATH, INCLUDE, LIBPATH, and LIB environment variables

instead of IDE paths for VC++ builds.

To attach the debugger from the command line, use:

VsJITDebugger.exe -p <pid>