



MSBuild

Sayed Ibrahim Hashimi – Jan 2008

www.sedodream.com

sayed.hashimi@gmail.com



What is MSBuild?

- Visual Studio build engine
- Builds all managed projects
- Built on XML
- Extensible & completely customizable by consumer (*you!*)
- Build doesn't depend on Visual Studio

MSBuild Sample

Sample MSBuild File (*Overview01.proj*)

```
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
```

```
<PropertyGroup>
  <Configuration>Release</Configuration>
</PropertyGroup>
```

Key / Value pairs

```
<ItemGroup>
  <Content Include="HelloWorld.proj"/>
</ItemGroup>
```

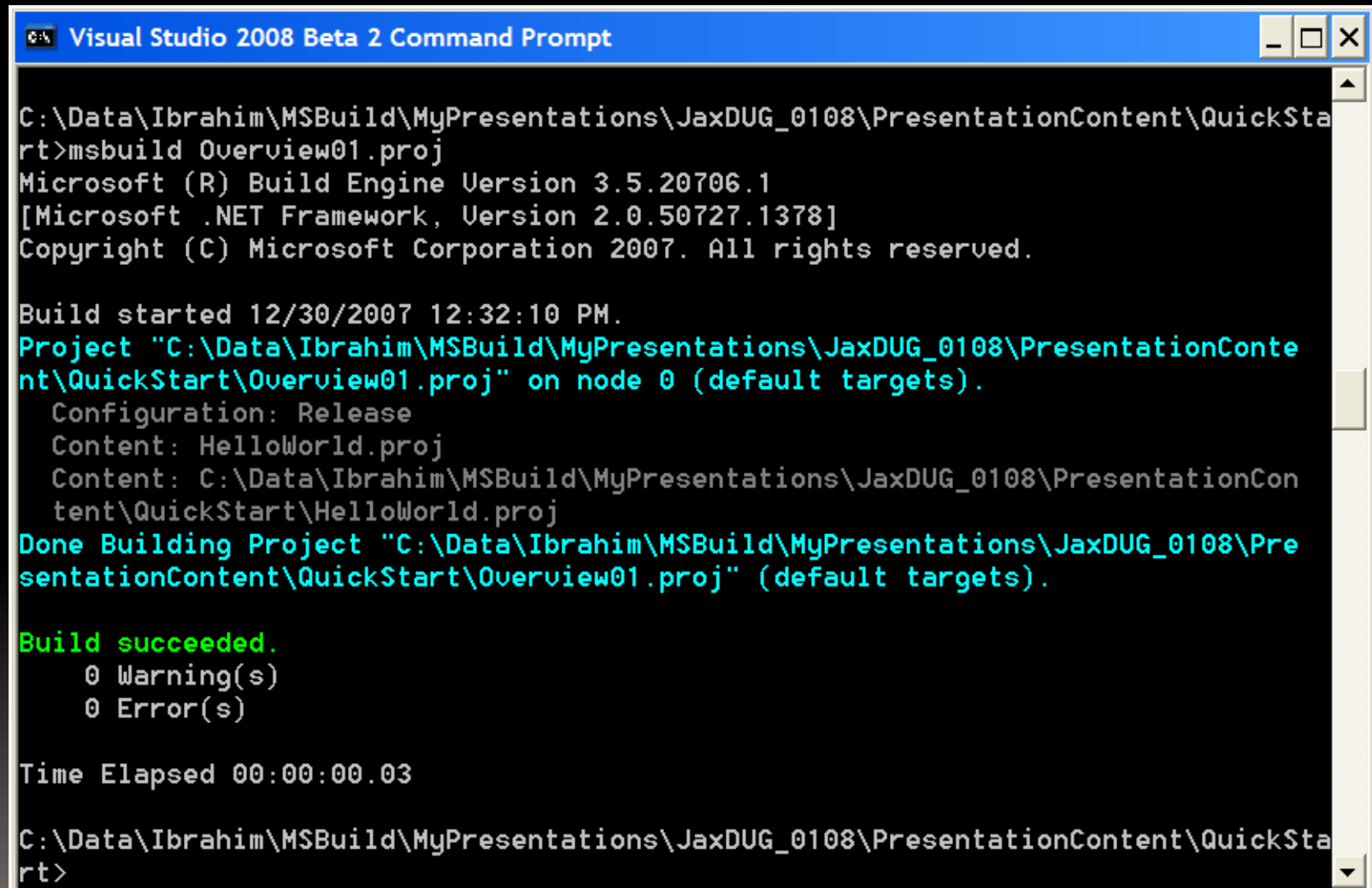
Usually refers to files

```
<Target Name="PrintContent">
  <Message Text="Configuration: $(Configuration)"/>
  <Message Text="Content: @(Content)"/>
  <Message Text="Content: @(Content->'%(FullPath)')"/>
</Target>
```

Targets consist of tasks

```
</Project>
```

MSBuild Sample Result



```
Visual Studio 2008 Beta 2 Command Prompt

C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\QuickStart>msbuild Overview01.proj
Microsoft (R) Build Engine Version 3.5.20706.1
[Microsoft .NET Framework, Version 2.0.50727.1378]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 12/30/2007 12:32:10 PM.
Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\QuickStart\Overview01.proj" on node 0 (default targets).
  Configuration: Release
  Content: HelloWorld.proj
  Content: C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\QuickStart\HelloWorld.proj
Done Building Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\QuickStart\Overview01.proj" (default targets).

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:00.03

C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\QuickStart>
```

Tasks Shipped with MSBuild

AL	GenerateBootstrapper	ResolveComReference
AspNetCompiler	GenerateDeploymentManifest	ResolveKeySource
AssignCulture	GenerateResource	ResolveNativeReference
CallTarget	GetAssemblyIdentity	SGen
CombinePath	GetFrameworkPath	SignFile
ConvertToAbsolutePath	GetFrameworkSdkPath	Touch
Copy	LC	UnregisterAssembly
CreateItem	MakeDir	Vbc
CreateProperty	Message	VCBUILD
Csc	MSBuild	Warning
Delete	ReadLinesFromFile	WriteLinesToFile
Error	RegisterAssembly	
Exec	RemoveDir	
FindUnderPath	RemoveDuplicates	
GenerateApplicationManifest	ResolveAssemblyReference	

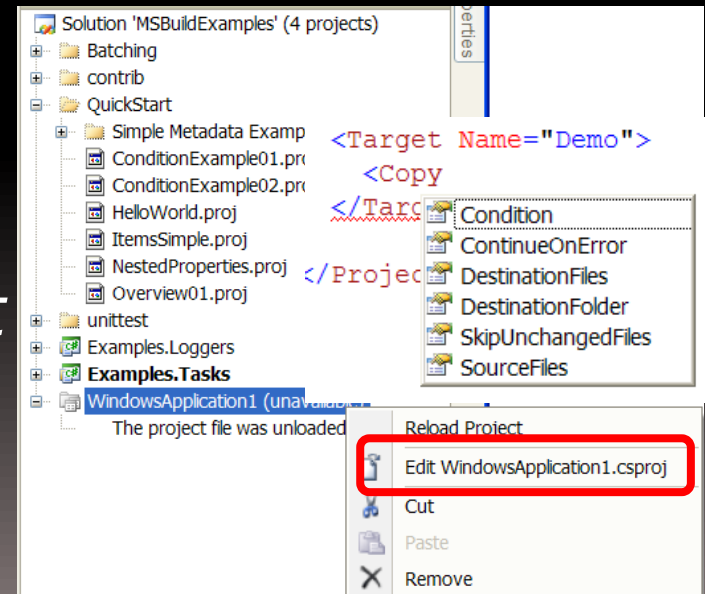
MSDN: <http://msdn2.microsoft.com/en-us/library/7z253716.aspx>

Use VS to Edit MSBuild Files

Visual Studio provides Intellisense on MSBuild files!

How to edit using Visual Studio

1. Right click on project in
2. Select Unload
3. Right click and pick *Edit*





Build Customization

3 Common Methods

- Legacy build events
 - Only for backwards compatibility
 - These are really properties in MSBuild
- Overriding pre-defined targets
 - Simplest suggested solution
- Extending the target dependencies
 - Most robust solution

Overridable Targets

These targets are declared specifically to be overridden.

BeforeBuild	AfterPublish
AfterBuild	BeforeResolveReferences
BeforeRebuild	AfterResolveReferences
AfterRebuild	BeforeCompile
BeforeClean	AfterCompile
AfterClean	BeforeResGen
BeforePublish	AfterResGen

More Info : <http://blogs.msdn.com/msbuild/archive/2005/11/23/496396.aspx>

Dependency Properties

These properties can be modified to extend the build process

CompileLicxFilesDependsOn	PublishOnlyDependsOn
CoreBuildDependsOn	RebuildDependsOn
CoreCleanDependsOn	ResGenDependsOn
CoreCompileDependsOn	ResolveAssemblyReferencesDependsOn
CreateManifestResourceNamesDependsOn	ResolveReferencesDependsOn
CreateSatelliteAssembliesDependsOn	RunDependsOn
GetRedistListsDependsOn	UnmanagedRegistrationDependsOn
PostBuildEventDependsOn	UnmanagedUnregistrationDependsOn
PreBuildEventDependsOn	BuiltProjectOutputGroupDependsOn
PrepareForBuildDependsOn	DebugSymbolsProjectOutputGroupDependsOn
PrepareForRunDependsOn	DocumentationProjectOutputGroupDependsOn
PrepareResourceNamesDependsOn	SatelliteDllsProjectOutputGroupDependsOn
PrepareResourcesDependsOn	SourceFilesProjectOutputGroupDependsOn
PublishBuildDependsOn	ContentFilesProjectOutputGroupDependsOn
PublishDependsOn	

More info : <http://blogs.msdn.com/msbuild/archive/2006/02/10/528822.aspx>

Build Customization - sample

```
<Project DefaultTargets="Build"
```

```
  xmlns="http://schemas.microsoft.com/build/2009"
  ToolsVersion="4.0.0"
  ...
```

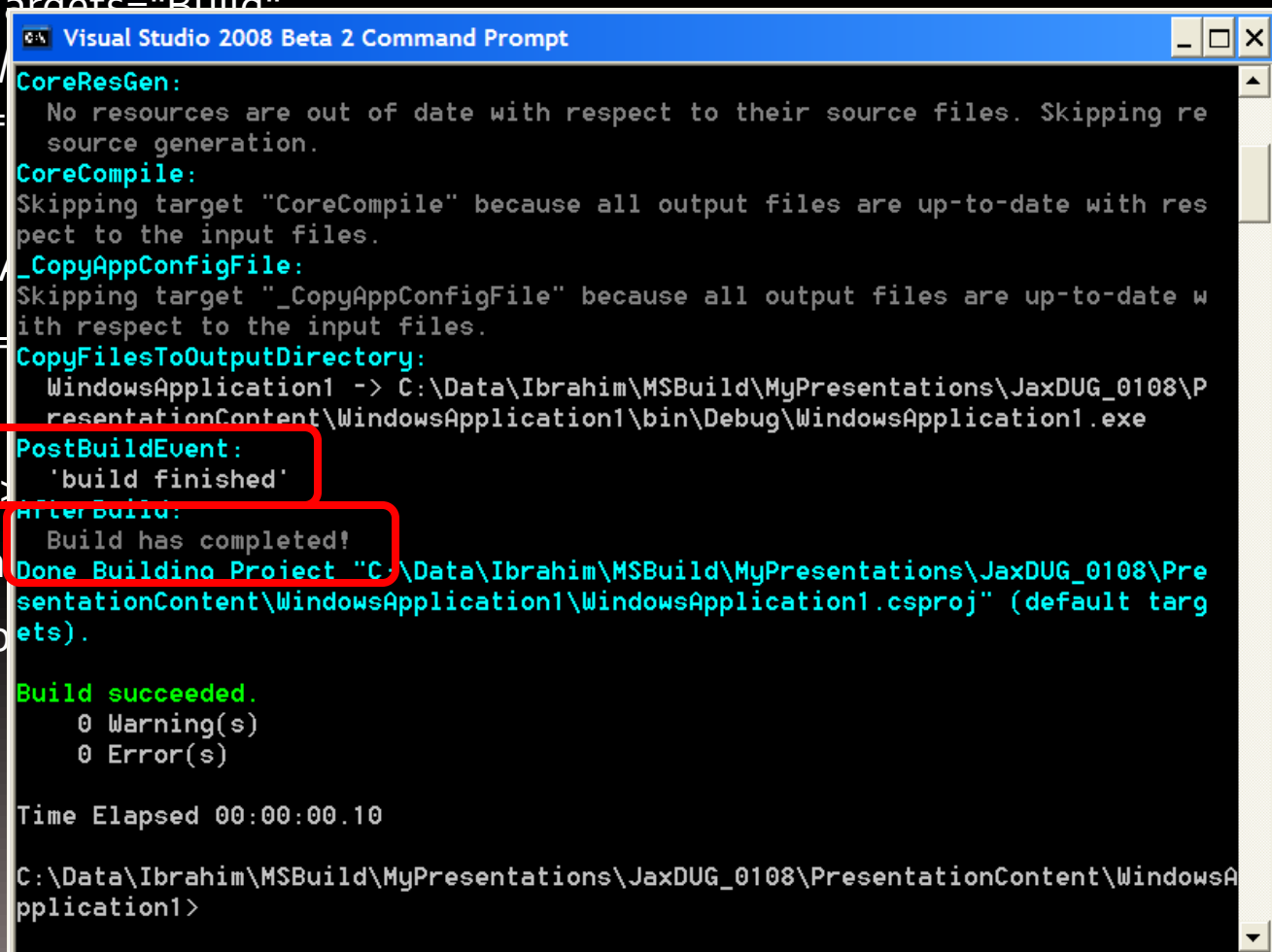
```
  <Target Name="Build"
    <Message Text="Building project"
    </Target>
```

```
  <PropertyGroup>
```

```
    <PostBuildEvent>
```

```
  </PropertyGroup>
```

```
</Project>
```

A screenshot of a Visual Studio 2008 Beta 2 Command Prompt window. The window title is "Visual Studio 2008 Beta 2 Command Prompt". The output shows the build process for a project named "JaxDUG_0108". The build is successful, with no warnings or errors. The output includes messages from CoreResGen, CoreCompile, and CopyFilesToOutputDirectory. A red box highlights the "PostBuildEvent" section, which shows the build has completed and the project has been built successfully. The output also shows the time elapsed (00:00:00.10) and the path to the output files.

```
Visual Studio 2008 Beta 2 Command Prompt
CoreResGen:
  No resources are out of date with respect to their source files. Skipping re
  source generation.
CoreCompile:
  Skipping target "CoreCompile" because all output files are up-to-date with res
  pect to the input files.
_CopyAppConfigFile:
  Skipping target "_CopyAppConfigFile" because all output files are up-to-date w
  ith respect to the input files.
CopyFilesToOutputDirectory:
  WindowsApplication1 -> C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\P
  resentationContent\WindowsApplication1\bin\Debug\WindowsApplication1.exe
PostBuildEvent:
  'build finished'
After Build:
  Build has completed!
Done Building Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\Pre
  sentationContent\WindowsApplication1\WindowsApplication1.csproj" (default targ
  ets).
Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:00.10

C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\WindowsA
  pplication1>
```

Build Customization – Sample 2

Target Dependency

Add to build dependency

```
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/build/2009" ToolsVersion="4.0">
```

...

```
<Import Project="$(MSBuildToolsPath)\Microsoft.Build.Tasks.v4.0.dll" />
```

```
<PropertyGroup>
```

```
  <BuildDependentTargets>
```

```
    $(BuildDependentTargets)
```

```
    CustomAfterBuild
```

```
  </BuildDependentTargets>
```

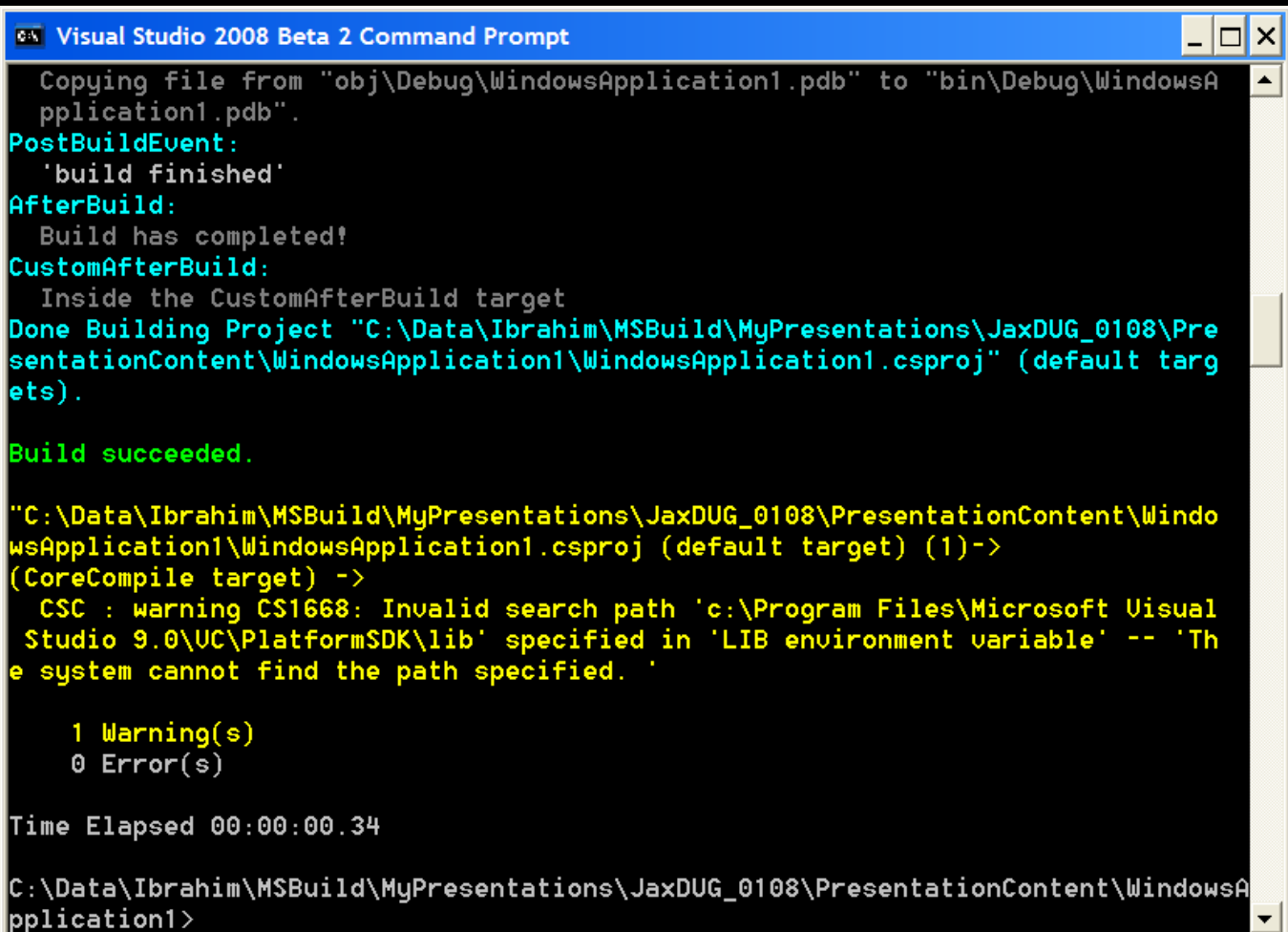
```
</PropertyGroup>
```

```
<Target Name="Build" />
```

```
  <Message Text="Build started" />
```

```
</Target>
```

```
</Project>
```



```
Visual Studio 2008 Beta 2 Command Prompt

Copying file from "obj\Debug\WindowsApplication1.pdb" to "bin\Debug\WindowsA
pplication1.pdb".
PostBuildEvent:
  'build finished'
AfterBuild:
  Build has completed!
CustomAfterBuild:
  Inside the CustomAfterBuild target
Done Building Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\Pre
sentationContent\WindowsApplication1\WindowsApplication1.csproj" (default targ
ets).
Build succeeded.
"C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Windo
wsApplication1\WindowsApplication1.csproj (default target) (1)->
(CoreCompile target) ->
  CSC : warning CS1668: Invalid search path 'c:\Program Files\Microsoft Visual
Studio 9.0\VC\PlatformSDK\lib' specified in 'LIB environment variable' -- 'Th
e system cannot find the path specified.'
1 Warning(s)
0 Error(s)

Time Elapsed 00:00:00.34

C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\WindowsA
pplication1>
```

Reserved Properties

- Predefined properties by MSBuild
- Cannot be overridden

MSBuildProjectDirectory	MSBuildBinPath
MSBuildProjectFile	MSBuildProjectDefaultTargets
MSBuildProjectExtension	MSBuildExtensionsPath
MSBuildProjectFullPath	MSBuildStartupDirectory
MSBuildProjectName	

More info : <http://msdn2.microsoft.com/en-us/library/ms164309.aspx>



MSBuild Items

- An item is a named group of objects
 - ▣ Typically points to files
- Items can be declared in 2 ways
 - ▣ Inside the ItemGroup element
 - ▣ Using the CreateItem task
- All items have metadata
 - ▣ Well known metadata (*next slide*)
 - ▣ Custom metadata

Well-known Metadata

Metadata that is available on every item

%(FullPath)	%(RecursiveDir)
%(RootDir)	%(Identity)
%(Filename)	%(ModifiedTime)
%(Extension)	%(CreatedTime)
%(RelativeDir)	%(AccessedTime)
%(Directory)	

```
<Target Name="CustomAfterBuild">
...

  <Zip Files="@(\OutputFiles->'%(FullPath)') "
        WorkingDirectory="$(OutputPath)"
        ZipFileName="$(OutputZipFileName)"/>
</Target>
```

Well-known Metadata

Metadata that is available on every item

%(FullPath)	%(RecursiveDir)
%(RootDir)	%(Identity)
%(Filename)	%(ModifiedTime)
%(Extension)	%(CreatedTime)
%(RelativeDir)	%(AccessedTime)
%(Directory)	

```
<Target Name="CustomAfterBuild">
...
  <Copy SourceFiles="@ (FilesToCopy)"
        DestinationFiles=
"@ (FilesToCopy)->'$(DestFolder)%(RecursiveDir)%(Filename)%(Extension)'"
        />
</Target>
```



Batching

- Very confusing
- Very useful
- Divides items into groups (batches)

2 Types of batching

- Task batching
- Target batching

Batching Sample 1

```
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <SourceFolder>$(MSBuildProjectDirectory)\src\</SourceFolder>
  </PropertyGroup>

  <ItemGroup>
    <SourceFiles Include="$(SourceFolder)*.txt"/>
  </ItemGroup>

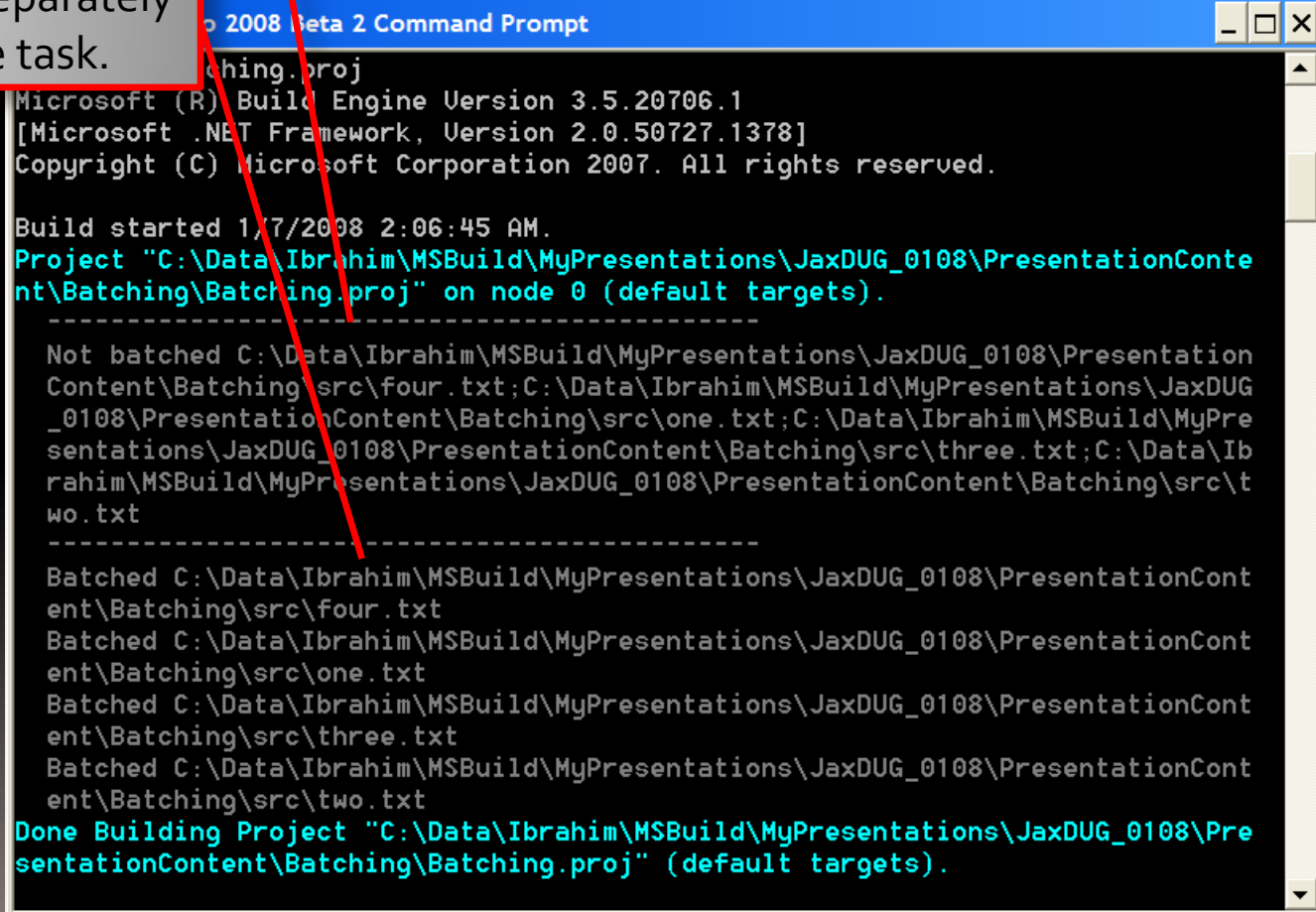
  <Target Name="TaskBatching">
    <Message Text="-----"/>
    <Message Text="Not batched @(SourceFiles->'%(FullPath)')"/>

    <Message Text="-----"/>
    <Message Text="Batched %(SourceFiles.FullPath)"/>
  </Target>
</Project>
```

Batching Sample 1

The items were converted into a single string & passed to the Message task.

Each file is passed separately into the Message task.



```
Microsoft (R) Build Engine Version 3.5.20706.1
[Microsoft .NET Framework, Version 2.0.50727.1378]
Copyright (C) Microsoft Corporation 2007. All rights reserved.

Build started 1/7/2008 2:06:45 AM.
Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\Batching.proj" on node 0 (default targets).
-----
Not batched C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\four.txt;C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\one.txt;C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\three.txt;C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\two.txt
-----
Batched C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\four.txt
Batched C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\one.txt
Batched C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\three.txt
Batched C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\two.txt
Done Building Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\Batching.proj" (default targets).
```

Batching Sample 2

```
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
...
  <ItemGroup>
    <SourceFiles Include="$(SourceFolder)one.txt">
      <Category>CatA</Category>
    </SourceFiles>
    <SourceFiles Include="$(SourceFolder)two.txt">
      <Category>CatB</Category>
    </SourceFiles>
    <SourceFiles Include="$(SourceFolder)three.txt">
      <Category>CatA</Category>
    </SourceFiles>
    <SourceFiles Include="$(SourceFolder)four.txt">
      <Category>CatB</Category>
    </SourceFiles>
  </ItemGroup>
...
</Project>
```

Batching Sample 2

```
<Target Name="TaskBatching">
  <Message Text="-----"/>
  <Message Text="Not batched @(SourceFiles->'%(Category)')"/>

  <Message Text="-----"/>
  <Message Text="Batched %(SourceFiles.Category)"/>
</Target>
```

```
Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationConte
nt\Batching\Batching2.proj" on node 0 (TaskBatching target(s)).
-----
Not batched CatA;CatB;CatA;CatB
-----
Batched CatA
Batched CatB
```

Batching Sample 2

```
<Target Name="TargetBatching" Outputs="% (SourceFiles.Category)">
  <Message Text="Category: %(SourceFiles.Category)"/>
  <Message Text="@ (SourceFiles)" />
</Target>
```

```
Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\Batching2.proj" on node 0 (TargetBatching target(s)).
  Category: CatA
  C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\one.txt;C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\three.txt
TargetBatching:
  Category: CatB
  C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\two.txt;C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\src\four.txt
Done Building Project "C:\Data\Ibrahim\MSBuild\MyPresentations\JaxDUG_0108\PresentationContent\Batching\Batching2.proj" (TargetBatching target(s)).
```

Compartmentalizing targets

- MSBuild files can import other files
 - Behavior & data can be included
 - Imported file should define steps
 - File importing should define data to act upon
 - Visual Studio uses this approach
 - *.csproj
 - Microsoft.csharp.targets
 - Microsoft.common.targets
- 

Reusable targets

4 Rules

1. Must be self contained
2. Transparent & extensible process
3. Overridable behavior
4. Defined contract

Contract defined in properties & items. Enforced by validation target.

```
<Target Name="ValidateNunitSettings">
  <!-- Validate assumptions that are contracted -->
  <Message Text="NunitAssemblies: @(NunitAssemblies)" Importance="low"/>

  <Error Condition="'$$(NunitOutputDir)'==' ' Text="NunitOutputDir property not defined"/>
  <Error Condition=" '@(NunitAssemblies)'==' ' Text="NunitAssemblies not defined"/>
  <Error Condition="'%(NunitAssemblies.ProjectName)'==' '
    Text="Atleast 1 item in NunitAssemblies doesn't have metadata 'ProjectName' defined."/>
  <Error Condition="!Exists('%(NunitAssemblies.FullPath)')" Text="Couldn't locate assembly at:
    %(NunitAssemblies.FullPath)" />
</Target>
```

Custom Tasks

Implement `Microsoft.Build.Framework.ITask`

- `Microsoft.Build.Utilities.Task`
- `Microsoft.Build.Utilities.ToolTask`

```
public class HelloWorld : Task
{
    public override bool Execute()
    {
        Log.LogMessageFromText("Hello MSBuild!",
            MessageImportance.High);
        return true;
    }
}
```


Task input/output

Inputs are properties with set

- Required attribute can be used

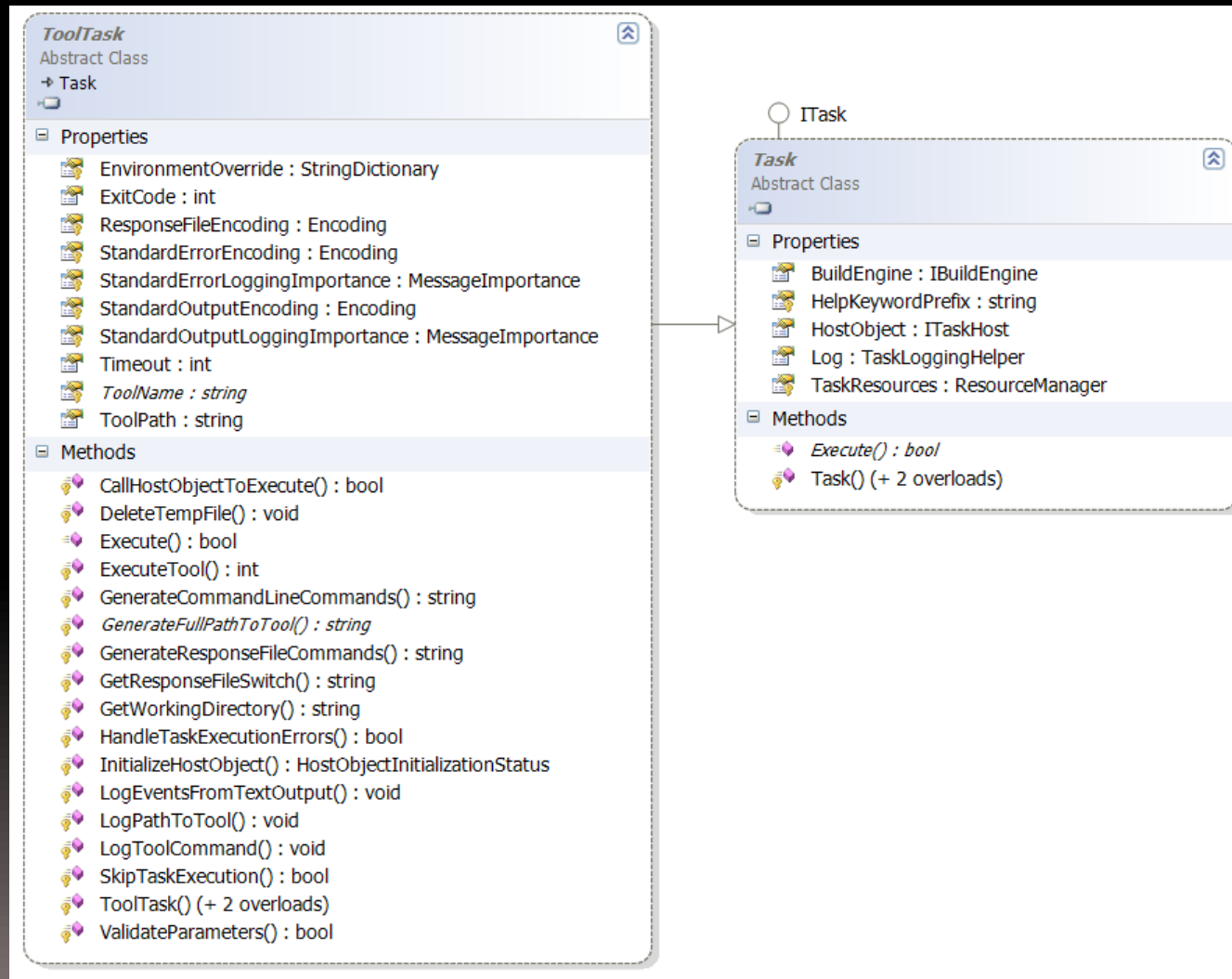
Outputs are properties with get

- Output attribute is required

```
[Required]
public ITaskItem[] SourceFiles
{
    get { return this.sourceFiles; }
    set { this.sourceFiles = value; }
}

[Output]
public ITaskItem[] MovedFiles
{
    get { return this.movedFiles; }
}
```

Task Class Diagram





Debugging tasks

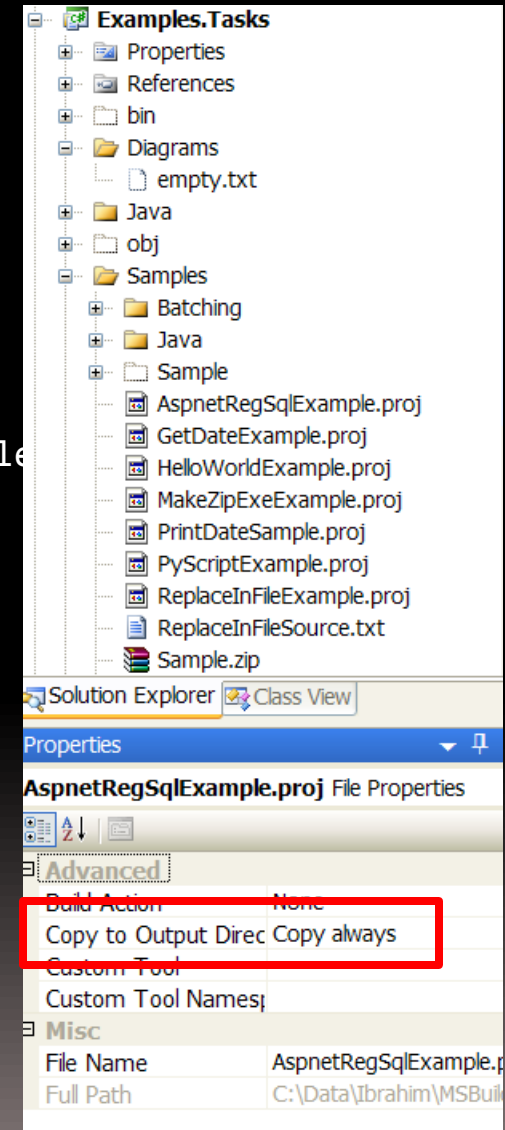
3 Common Methods

1. Output messages to log
 - ▣ Most primitive
2. Insert sleep, attach Visual Studio
 - ▣ You either modify code or MSBuild file
3. Configure VS to start msbuild.exe debug
 - ▣ No changes required
 - ▣ Supports Edit & Continue

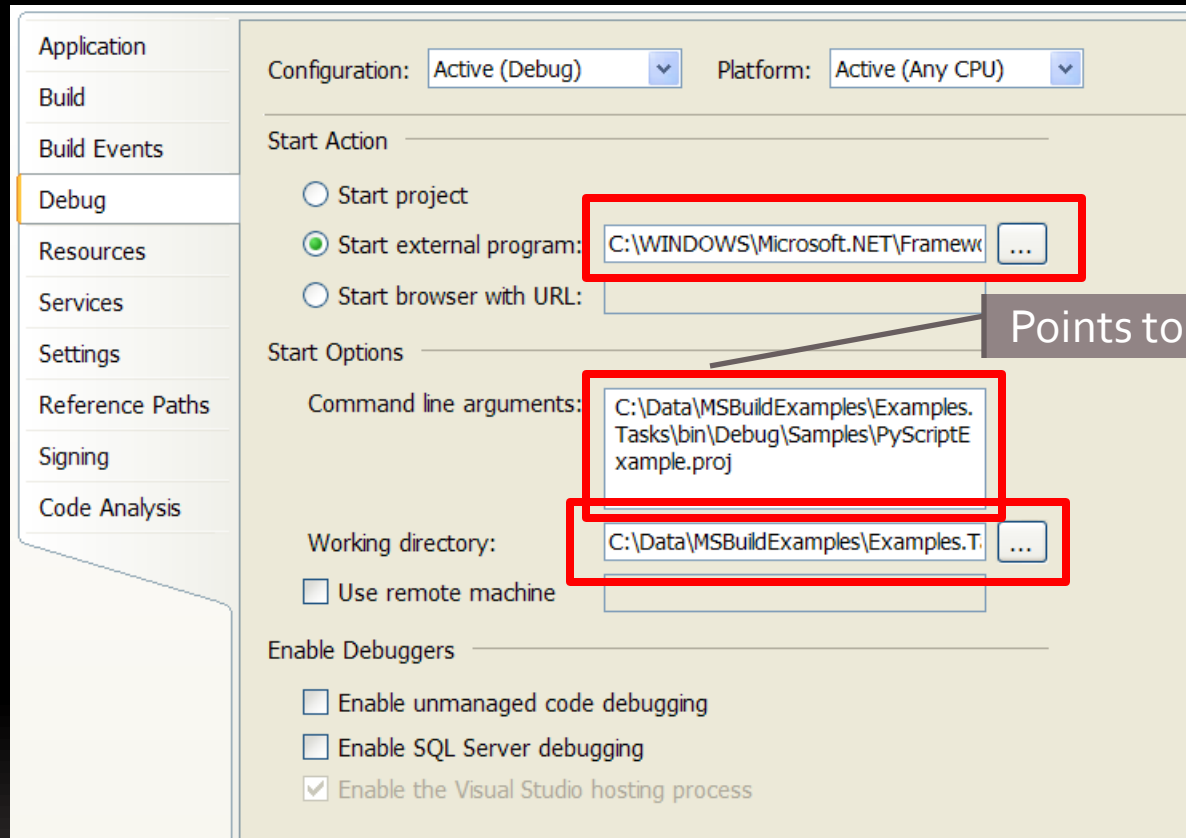
Debugging with Visual Studio

Create test scripts in project
Set files to copy to output directory
In your sample use task built

```
<UsingTask  
  AssemblyFile="$(MSBuildProjectDirectory)\..\Example  
  TaskName="ReplaceInFile"/>
```



Debugging with Visual Studio



Points to script in output directory

Open Source Tasks

msbuildtasks.tigris.org

- Over 50 tasks

FileUpload	Sleep
FxCop	SqlExecute
NUnit	Svn related tasks
NDoc	VSS related tasks
Zip	IIS related tasks

codeplex.com/sdctasks

- Over 300 tasks!
- Created by Microsoft UK

CreateGuid	Tfs tasks
Msi.Install	Source safe
CreateKey	Sql server
DeleteKey	FTP
Source Depot	Wix

codeplex.com/sedodream

- Over 15 tasks

ReplaceInFile
ReplaceInText
TempFile
GetDate



Resources

[Inside MSBuild](#) : MSDN 06/2006

[WiX Tricks](#) : MSDN 03/2007

Deploying .NET Applications (Apress 05/2006)

My blog : www.sedodream.com

MSBuild Team Blog : blogs.msdn.com/msbuild

[MSBuild Forum](#)

[MSDN Reference](#)