

NAME

xbuild – Mono's tool to build MSBuild project files

SYNOPSIS

xbuild [option] [project-file]

DESCRIPTION

xbuild can be used to build MSBuild project files. Integrated Development Environments like MonoDevelop and "Microsoft Visual Studio .NET" use msbuild project file format.

xbuild takes the path of the project or solution file to build, as the main argument. If no file is specified then it tries to build any solution file or project file in the current directory, that has a *.proj extension.

OPTIONS

/target:T1[,TN]

List of targets to build

/property:Name=Value

Set or override project properties

/logger:<logger>

Custom logger to log build events

/toolsversion:version , /tv:version

Specify the toolset version to use.

Eg. /tv:4.0

This would cause 4.0 version of the Microsoft.Common.targets (among others) to be used. This overrides

the value specified in a project file.

Valid values: 2.0, 3.0, 3.5 and 4.0

/verbosity:<level>

Logger verbosity level : quiet, minimal, normal, detailed, diagnostic

/validate

Validate the project file against the schema

/validate:<schema>

Validate the project file against the specified schema

/consoleloggerparameters:<params> , /clp:<params>

Parameters for the console logger : PerformanceSummary, Summary, NoSummary, NoItemAndPropertyList, Verbosity

/filelogger[n] , /fl[n]

Logs the output to a file, named 'msbuild.log' by default. If the optional number 'n' is specified, then it logs to 'msbuild[n].log'. Parameters for this logger, including the log file name can be specified via a corresponding /flp[n] option. Default verbosity for file loggers is 'detailed'. 'n' can be between 1-9.

/fileloggerparameters[n]:<params> , /flp[n]:<params>

Parameters for a file logger. This implies a corresponding /fl[n]. 'n' can be between 1-9, and is used to add upto 10 file loggers. Parameters can be, besides the ones available for console logger:

LogFile=<file>

File to which the build log will be written.

Append

If this is used, then the log file is appended to, else a new one will be created.

Encoding=<encoding>

Encoding to use for the log file, eg. UTF-8, ASCII.

Eg: xbuild foo.csproj /flp:Verbosity=normal "/flp1:LogFile=build.log;Append;Encoding=ASCII"

/nologo

Don't show the initial xbuild banner

/help

Show xbuild usage

/version

Display xbuild version

ENVIRONMENT VARIABLES

XBUILD_EMIT_SOLUTION

If this variable is set, then the project file generated from a solution file is emitted.

XBUILD_LOG_REFERENCE_RESOLVER

References are resolved by trying a list of assembly search paths (\$(AssemblySearchPaths)). If xbuild is unable to resolve a reference, then it logs details of why the various search paths failed. If this variable is set, then it logs the same even for references that were resolved successfully. These logs show up if the verbosity is set to detailed or higher.

MSBuildExtensionsPath

MSBuild extensions are usually installed in \$(MSBuildExtensionsPath), which xbuild resolves to \$prefix/lib/mono/xbuild . When used in Import, like:

```
<Import Project="$(MSBuildExtensionsPath)\TestTargets.targets"/>
```

xbuild tries various values for the msbuild property \$(MSBuildExtensionsPath), in order:

1. Paths specified in the environment variable \$MSBuildExtensionsPath.
2. /Library/Frameworks/Mono.framework/External/xbuild on Mac OSX.
3. \$XDG_CONFIG_HOME/xbuild/tasks (or Environment.SpecialFolder.ApplicationData)
4. \$prefix/lib/mono/xbuild (default location)

Anywhere else in the project files, \$(MSBuildExtensionsPath) will always resolve to the default location. This is a xbuild-only feature. This is also applicable for the properties \$(MSBuildExtensionsPath32) and \$(MSBuildExtensionsPath64), and the environment variables have the corresponding names - MSBuildExtensionsPath32/64 .

XBUILD_FRAMEWORK_FOLDERS_PATH

With ToolsVersion 4.0, projects can target arbitrary frameworks referenced by TargetFramework-Moniker, which is of the format:

Identifier,Version=<version>[,Profile=<profile>]

Eg. ".NETFramework,Version=v4.0"

The 3 parts of the moniker are given by the msbuild properties:

\$(TargetFrameworkIdentifier), \$(TargetFrameworkVersion) and \$(TargetFrameworkProfile)

This moniker maps to a framework description file on disk:

```
<framework_root>/Identifier/Version/[Profile]/RedistList/FrameworkList.xml
```

This file is used to determine the path where to find the framework assemblies for this particular framework.

Framework root here is configurable and is resolved in the following order:

1. Paths specified in the environment variable `$XBUILD_FRAMEWORK_FOLDERS_PATH`
2. `/Library/Frameworks/Mono.framework/External/xbuild-frameworks` on Mac OSX.
3. MSBuild property `$(TargetFrameworkRoot)`
4. `$prefix/lib/mono/xbuild-frameworks` (default location)

XBuild tries the paths given above, in order, till it finds a `FrameworkList.xml` for the moniker. Running with `/v:detailed` or higher verbosity will show the various paths that it tries.

The `FrameworkList.xml` itself just has a root element like:

```
<FileList Name=".NET Framework 3.5" TargetFrameworkDirectory="..\..\..\3.5" Include-
Framework="v3.0">
</FileList>
```

Here the `TargetFrameworkDirectory` attribute specifies the directory where the assemblies for this particular framework can be found. If this is not set, then the parent of the folder containing the xml file is taken as the framework directory.

`IncludeFramework` attribute specifies the version of a framework (under the **same** `$(TargetFrameworkIdentifier)`) which should be included in the final list of Target framework directories.

XBUILD_COLORS

If this variable is set, it contains a string of the form `"type=foreground,type=.."` that specifies which color to use to display errors/warnings etc on some terminals. Type here can be:

errors, warnings, messages or events

events: These are project/target/task start and end event messages.

The possible colors for foreground are: **black, red, brightred, green, brightgreen, yellow, brightyellow,** blue, brightblue, magenta, brightmagenta, cyan, brightcyan, grey, white and brightwhite.

For example, you could set this variable from your shell:

```
export XBUILD_COLORS
XBUILD_COLORS=errors=brightred,warnings=blue
```

You can disable the built-in color scheme by setting this variable to "disable".

NOTES

On Mac OSX, for locating `pkg-config` (.pc) files, xbuild additionally looks in `/Library/Frameworks/Mono.framework/External/pkgconfig` .

SEE ALSO

mono(1), **mcs(1)**

COPYRIGHT

Copyright (C) 2011 Novell, Inc (<http://www.novell.com>)

MAILING LISTS

Visit <http://lists.ximian.com/mailman/listinfo/mono-devel-list> for details.

WEB SITE

Visit: <http://www.mono-project.com> for details