Building Solutions and Projects

Many dotTEST features require source code to be compiled into binaries. By default, dotTEST attempts to build solutions and projects prior to analyzing them.

Delegating the Build to MSBuild or Visual Studio

dotTEST delegates the build to `msbuild.exe` by default when Visual Studio is not installed on the machine. If an appropriate version of Visual Studio is present, the build is performed by the `devenv.exe /Build` mechanism.

Set the `dottest.build.builder_id` property to either `visualstudio` or `msbuild` to explicitly set Visual Studio or MSBuild as the builder. See Base Configuration Settings for details.

Depending on Pre-built Code

If your code is compiled prior to analysis, you can skip the building phase during analysis by passing the `-nobuild` command line switch or setting the `dottest.build.nobuild` property to `true` (see Base Configuration Settings for details).

If you use the NuGet mechanism to process packages in your project, they are automatically restored before the project is built. If the building phase is skipped, NuGet packages are not restored.

The default timeout for restoring packages is 360 seconds and can be changed with the `dottest.package.restore_timeout` option (see Base Configuration Settings for details).

Specifying Solution Configuration and Target Platform

If your code is built during or prior to analysis, dotTEST needs to know the correct Solution Configuration (e.g. Debug, Release, or other) and Target Platform (e.g. Any CPU, x86, or other). dotTEST automatically attempts to choose the most suitable Configuration and Platform, but it is recommended to specify them explicitly.

Use the `-solutionConfig` and `-targetPlatform` command line switches to specify the Solution Configuration and Target Platform; see Command Line Options for details.

Verifying the Required Build Artifacts

Prior to analysis, dotTEST needs to verify that all required build artifacts, such as .exe, .dll or .pdb files, are available. Artifacts may not be available due to build issues or incorrect setup, which may prevent dotTEST from performing analysis or collecting complete analysis results.

If dotTEST is unable to find all the required .exe, .dll and .pdb files, the missing artifacts will be listed in the Setup Problems section of your report.

Restoring Packages Before the Build

If you use NuGet to process packages in your project, dotTEST automatically restores the packages before the project is built by executing the `nuget restore` command. The default timeout for restoring packages is 360 seconds and can be changed with the `dottest.package.restore_timeout` option (see Base Configuration Settings for details).