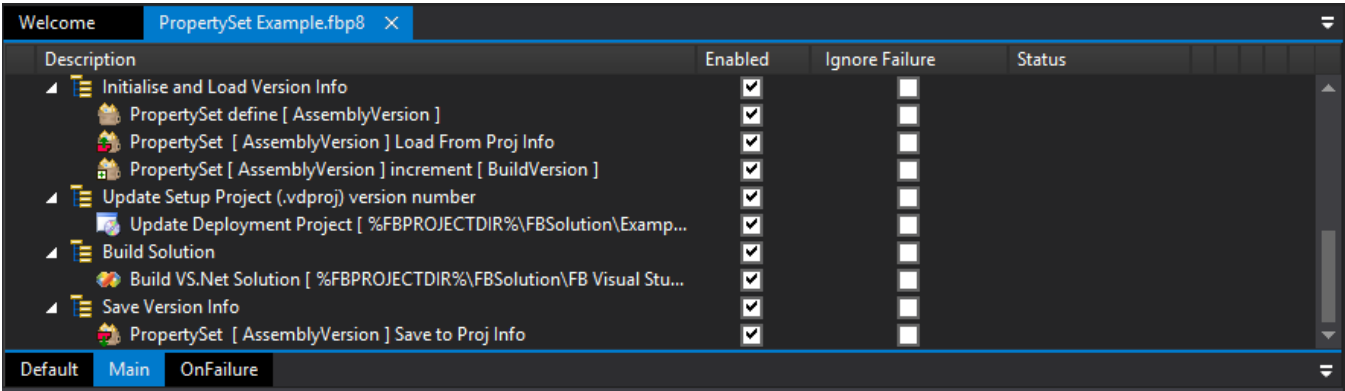


# FinalBuilder Basic Concepts

## Actions and Targets

FinalBuilder projects have one or more [Targets](#), which in turn contain Actions to perform the build. Each action represents one step in the process, by performing a task such as copying files, creating directories, etc.



## Default Target

By default each FinalBuilder project has a Default Target. The project starts at the first enabled action in the Default Target. If an error occurs then it can be handled by [Try Catch Actions](#). If not handle the build will terminate reporting the error which caused it to terminate. In a change from previous versions of FinalBuilder the developer of the build script is responsible for handling all error conditions as required. Previous versions of FinalBuilder had a global error handler Target which would be caused if an unhandled error caused the build to stop.

## User Targets

In addition to the Default Target, FinalBuilder also allows you to define additional Targets for each project. These can be called with the [Run Target Action](#). Targets are ideal for grouping sets of actions which are to be performed as a group, or called multiple times with in the script. The greatest benefit of FinalBuilder is the ability to call the command line execution with the list of targets required to be performed for the current run of the script. This means Targets such as Clean, Build, Rebuild, Deploy, RunTests, and BuildHelp could all be created in the one script and each picked as required for the build required at any future point in time.

Targets can also define dependencies, so for example a Build target could depend on a Clean Target - when the Build target is invoked, the Clean Target will automatically be invoked before Build.

## See Also

[Actions Tab](#) | [Targets](#)