

[Contact](#) | [Licen](#)[Home](#)[Features](#)[Status](#)[Documentation](#)[Download](#)

# Quick Start Guide

## Welcome

Welcome to FASTBuild!

This guide should help you quickly get upto speed with the basics of FASTBuild.

NOTE: This guide glosses over a lot of the subtle details of the configuration language and what is possible with FASTBuild. For a detailed guide to the configuration language, see the [syntax guide](#).

## Running FASTBuild

FASTBuild is a stand alone executable that does not need to be installed. It can be placed anywhere on your system. It simply needs to be in the path you wish to compile from. Simply unpack the "fbuild.exe" of your choice (64bit or 32bit) anywhere you like, and ensure it's in the environment path, or the current directory for the command prompt.

To verify everything is correct, open a command prompt, and type "fbuild -version". You should see something like the following:

```
c:\p4\Code>fbuild -version
FBuild - v0.50 (x86)

c:\p4\Code>
```

You're now ready to start configuring FASTBuild.

## Configuration File

FASTBuild expects a configuration file called 'fbuild.bff' in the current directory. This text file contains the configuration of all build targets, defined using a syntax which is similar in style to C.

The fbuild.bff file is where all build configuration targets are defined. Targets should be defined in dependency order. In this quick guide, everything is fixed for simplicity. More realistic build configurations will contain loop constructs and variables to avoid duplication of configuration information. (See the detailed [syntax guide](#) for more information).

## Defining Targets

You can define some libraries in the fbuild.bff for FASTBuild to build as follows:

```
.Compiler          = 'Bin\VC\cl.exe'
.Librarian          = 'Bin\VC\lib.exe'
.CompilerOptions    = '%1 /Fo%2 /c /Z7'
.LibrarianOptions   = '/NODEFAULTLIB /OUT:%2 %1'

Library( 'libA' )
{
    .CompilerInputPath = 'Src\LibA\'
    .CompilerOutputPath= 'Out\LibA\'
    .LibrarianOutput   = 'Out\LibA\libA.lib'
}
Library( 'libB' )
{
    .CompilerInputPath = 'Src\LibB\'
    .CompilerOutputPath= 'Out\LibB\'
    .LibrarianOutput   = 'Out\LibB\libB.lib'
}
```

"Library" is one of several functions supported by FASTBuild. In the example above, some properties of the Library function are set outside the scope of the function to avoid duplication. This pattern is used a lot in FASTBuild configurations in order to reduce configuration complexity.

In function properties, %1 and %2 are special tokens which are replaced at build time with context dependent values. In this case, the compiler and linker input and output

files. See the [Syntax Guide](#) and [Function Reference](#) for more details.

## Building Targets

Once targets are defined in the fbuild.bff, they can be built from the command line. You can build a target by specifying it on the fbuild command line as follows:

```
c:\p4\Code>fbuild Out\LibA\LibA.lib
```

"Library" is one of several functions that creates an alias for your target, so the following is also valid (as defined in the example above):

```
c:\p4\Code>fbuild libA
```

Naturally aliases are used a lot to simplify invoking builds of particular targets.

© 2012-2017 Franta Fulin