



## GUIDE BOT FOR T3D INSTALLATION INSTRUCTIONS

This document will help to install *Guide Bot* software into an existing *Torque 3d* based project.

**Before performing any merging make sure you have a backup copy of all of your source files!**

Merging process consists of the following steps:

1. Choosing appropriate *Guide Bot* distribution (GMK or non-GMK).
2. Merging C++ source files.
3. Adding *Guide Bot*'s C++ files to your project file (\*.vcproj).
4. Adding *Guide Bot*'s library to your T3D solution.
5. Updating scripts and data files.

### 1. Choosing appropriate Guide Bot distribution

There are two *Guide Bot* distributions available: for owners of [GMK add-on](#) and for ordinary T3D users.

Owners of binary distribution of T3D can use *Guide Bot* as well, they should jump to straight to step 5.

**If you are GMK owner you should install *Guide Bot* over the version of T3D patched with the latest version of GMK!**

## 2. Merging C++ (\*.h, \*.cpp) source files

*Guide Bot* consists of two main parts: core library and T3D integration layer. In this step we will add integration layer files to the existing T3D project.

If you want to start a new *Guide Bot* based project, you will just need to copy files from *Guide Bot* package to your T3D sources folder (probably it has path like this “C:\Torque 3D Pro\engine\sources”) and overwrite all files with the same name.

Owners of GMK should copy files from “[guideBotT3D\engine\sources](#)” while other users “[guideBotT3D\engine\sourcesNoGmk](#)”.

If you have modified original engine code by yourself you should merge your code with *Guide Bot*. To simplify merging process you can use a CodePatch utility.

You can download CodePatch from [this page](#).

CodePatch is using three-way merge (also know as merging from a common ancestor) algorithm to merge text files. CodePatch compares files from three folders each one contains a different modification of the engine. The first one is your current engine version (should be specified by “**my path**”). The second folder is the unmodified source code from the original Torque installation (“**base path**”, probably will looks like this “c:\Torque 3D Pro\Engine\”). If you have overwritten sources with your own modifications you’ll need to install Torque engine to new folder and provide CodePatch with appropriate path to original engine sources. Third is the folder with sources of *Guide Bot* (“**their path**”). CodePatch will compare files from “my” path and “their” relatively to “base” and place merged file in the result folder (specified by “**result path**”).

Some times CodePatch will not be able to merge your code automatically. This situation is known as a *conflict*. It happens when changes in files from “my” and “their” folders contradict to each other. By default CodePatch will simply put both changes in result file for you to handle it manually later. In this case you should look for '<<<<<<<' string (conflict descriptions starts with that) in result sources.

All *Guide Bot* changes in code are commented with special marks, so you can easily identify such blocks for example:

```
//.logicking guideBot >>
namespace GuideBot { class WorldObject; }
//.logicking guideBot <<
```

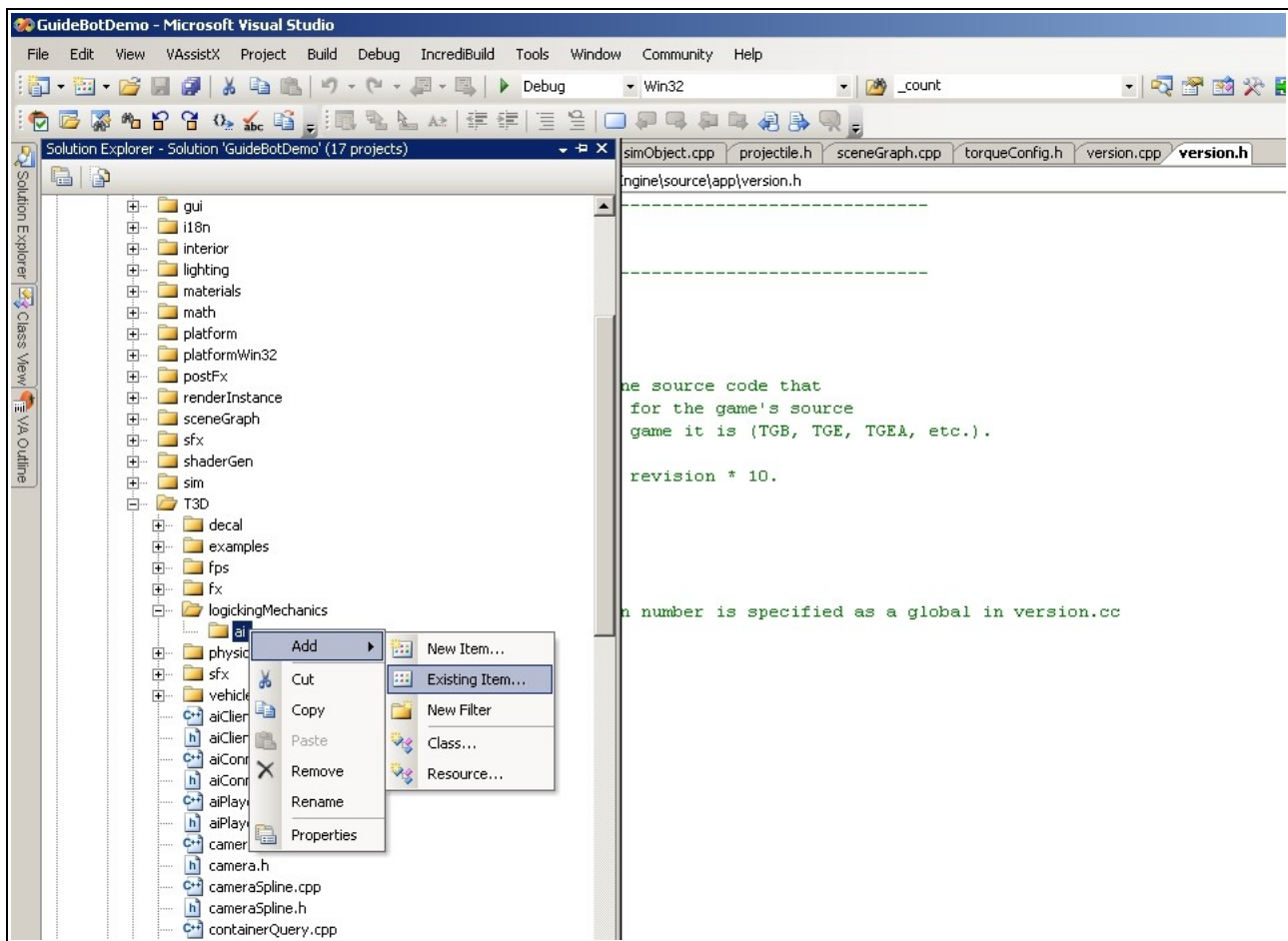
The other way to handle conflicts is to set “**Use external program to resolve conflicts**” checkbox and provide CodePatch with path to the WinMerge in the “**Program path**” field.

## 3. Adding Guide Bot's C++ files to your project file

To finish the integration of *Guide Bot* C++ code you should make changes to your Visual Studio project file. Actually you have to add *Guide Bot* \*.cpp and \*.h files to your project.

To add file to the C++ project you should right-click on the main **T3D DLL** (it might be called in a different way) project in your solution explorer and choose *Add->Existing Item*, just like show on

the image below.



Here's list of files to add:

```
engine\source\T3D\logickingMechanics\guideBot\enhancedPlayer.cpp
engine\source\T3D\logickingMechanics\guideBot\enhancedPlayer.h
engine\source\T3D\logickingMechanics\guideBot\guideBotConsoleMethods.cpp
engine\source\T3D\logickingMechanics\guideBot\sceneWorldObject.cpp
engine\source\T3D\logickingMechanics\guideBot\sceneWorldObject.h
engine\source\T3D\logickingMechanics\guideBot\scriptAction.cpp
engine\source\T3D\logickingMechanics\guideBot\scriptAction.h
engine\source\T3D\logickingMechanics\guideBot\torqueGuideBotPlatform.cpp
engine\source\T3D\logickingMechanics\guideBot\torqueGuideBotPlatform.h
```

#### 4. Adding *Guide Bot*'s library to your T3D solution.

In order to compile you should add

##### **Step 1**

Copy “*guideBotT3D\engine\lib\guideBot*” directory to “*Torque 3D\engine\lib*” directory.

Now you will need to change properties of your project in Visual Studio. You can access properties by pressing Alt+F7.

##### **Step 2**

Provide path to additional library directories.

*Project Properties->Linker->General->Additional Library Directories*

*../../../../engine/lib/guideBot/lib*

##### **Step 3**

Provide names of libraries files to link.

*Project Properties->Linker->Input->Additional Dependencies*

debug: *guideBot\_d.lib*

release: *guideBot.lib*

##### **Step 4**

Provide path to include files.

*Project Properties->C/C++->General->Additional Include Directories*

**Bullet**

*../../../../engine/lib/guideBot/include*

##### **Step 5**

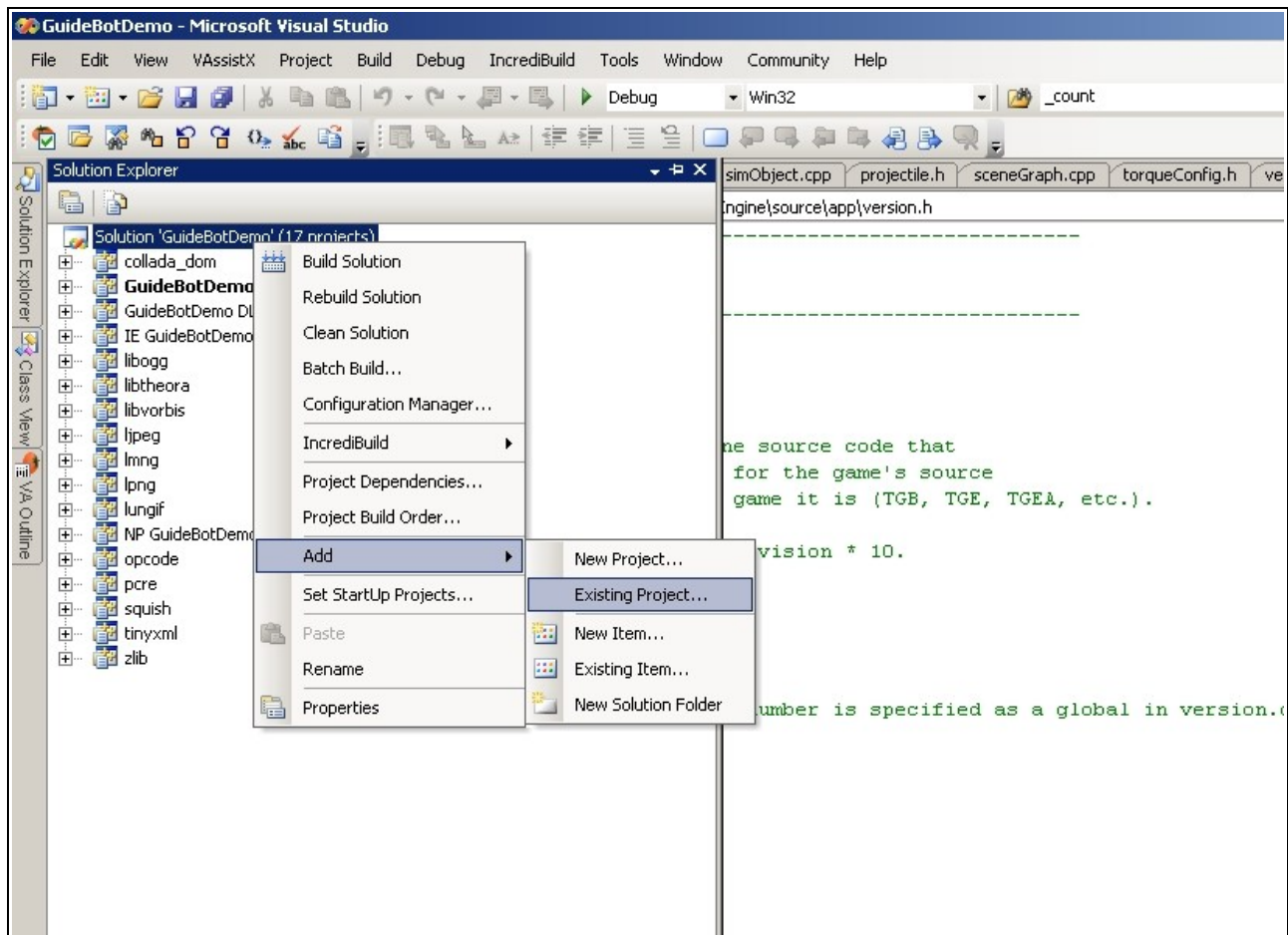
Rebuild your project in your VC++.

##### **Step 6**

If you want to you can add source files of *Guide Bot* library to you solution as well.

To add file to the C++ project you should right-click on the main **T3D DLL** (it might be called in a different way) project in your solution explorer and choose *Add->Existing Project*, and select file:

*Torque 3D/engine/lib/guideBot/guideBot.vcproj*



## 5. Updating scripts and data files

Owners of GMK should copy files from [“guideBotT3D\game”](#) , while other users should first copy [“guideBotT3D\game”](#) and then [“guideBotT3D\gameNoGmk”](#) and overwrite all files.

Updating scripts are nearly identical to merging source code. If you’re starting a brand new project you only have to overwrite files from “game” folder of your project with the ones of *Guide Bot*. Otherwise CodePatch will be in use again. This time you should provide paths to the “game” folder in your project, the original base project, and game folder within *Guide Bot* to “my path”, “base path” and “their path” respectively.

The installation process is over now. If every thing has gone smoothly you should be able to compile your engine from source codes and play with *Guide Bot* features and editor.