



# Getting Started with Embedded Software

---

## Summary

This tutorial shows how to create an embedded software project with Altium Designer.

Tutorial

TU0122 (v1.2) June 08, 2005

---

This tutorial presumes you are familiar with programming in C/assembly and have basic knowledge of embedded programming. It contains an overview of the TASKING tools available in Altium Designer. It describes how you can add, create and edit source files in an embedded project and how to build an embedded application.

The example used in this tutorial is a Hello World program in C. Other examples are supplied in the `\Program Files\Altium2004\Examples\FPGA Processors` folder.

## Embedded Software Tools

---

With the TASKING embedded software tools in Altium Designer you can write, compile, assemble and link applications for several targets, such as the TSK51x/TSK52x, TSK80x, TSK165x, PowerPC and TSK3000. Figure 1 shows all components of the TASKING toolchain with their input and output files.

The C compiler, assembler and debugger are target dependent, whereas the linker and the librarian are target independent. The **bold** names in the figure above are the executable names of the tools. Substitute *target* with one of the supported target names, for example, **c51** is the TSK51x/TSK52x C compiler, **cz80** is the TSK80x C compiler, **as165x** is the TSK165x assembler, etc.

## Getting Started with Embedded Software

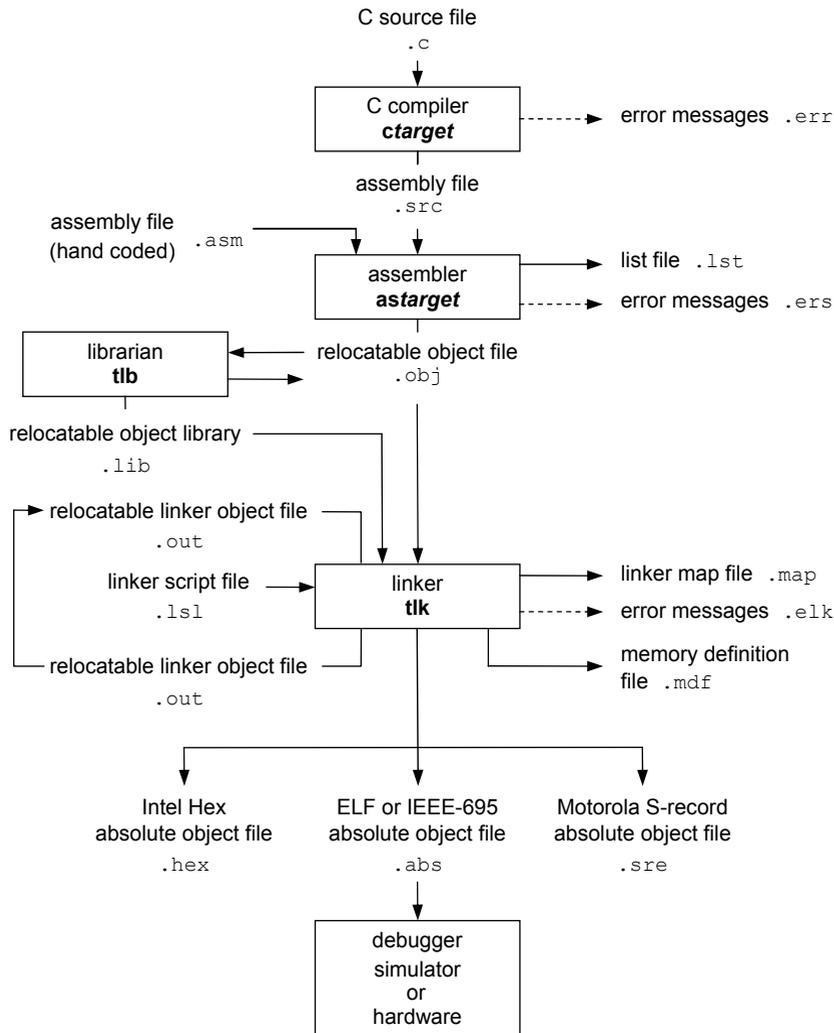


Figure 1: Toolchain overview

The following table lists the file types used by the TASKING toolchain.

Extension	Description
<b>Source files</b>	
.c	C source file, input for the C compiler
.asm	Assembler source file, hand coded
.lsl	Linker script file
<b>Generated source files</b>	
.src	Assembler source file, generated by the C compiler, does not contain macros

Extension	Description
<b>Object files</b>	
.obj	Relocatable object file, generated by the assembler
.lib	Archive with object files
.out	Relocatable linker output file
.abs	IEEE-695 or ELF/DWARF 2 absolute object file, generated by the locating part of the linker
.hex	Absolute Intel Hex object file
.sre	Absolute Motorola S-record object file
<b>List files</b>	
.lst	Assembler list file
.map	Linker map file
.mcr	MISRA C report file
.mdf	Memory definition file
<b>Error list files</b>	
.err	Compiler error messages file
.ers	Assembler error messages file
.elk	Linker error messages file

Table 1: File extensions

## Creating an Embedded Project

To start working with Altium Designer, you first need a project. A project makes managing your source documents and any generated outputs much easier. For embedded software, you need to create an Embedded Software project.

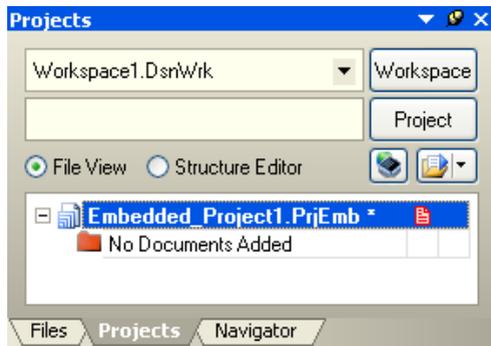
To create a new Embedded Software project:

1. Select **File » New » Project » Embedded Project** from the menu, or click on **Blank Project (Embedded)** in the **New** section of the **Files** panel. If this panel is not displayed, click on the **Files** tab at the bottom of the Design Manager panel.

Alternatively, you could click on **Embedded Software Development** in the **Pick a task** section of the Altium Designer Home Page (**View » Home**) and then click on **New Blank Embedded Software Project**.

2. The **Projects** panel opens. The new project file, `Embedded Project1.PrjEmb`, is listed here with no documents added.

## Getting Started with Embedded Software



3. Rename the new project file (with a `.PrjEmb` extension) by selecting **File » Save Project As**. Navigate to a location where you want to save the project on your hard disk, type the name `GettingStarted.PrjEmb` in the File name field and click on **Save**.

### Adding a new source file to the project

If you want to add a new source file (C or assembly or text file) to your project, proceed as follows:

1. In the **Projects** panel, right-click on `GettingStarted.PrjEmb` and select **Add New to Project » C File**. A new C source file, `Source1.C`, is added to the Embedded Software project under the folder named `Source Documents` in the **Projects** panel. The Text Editor opens ready for your input.

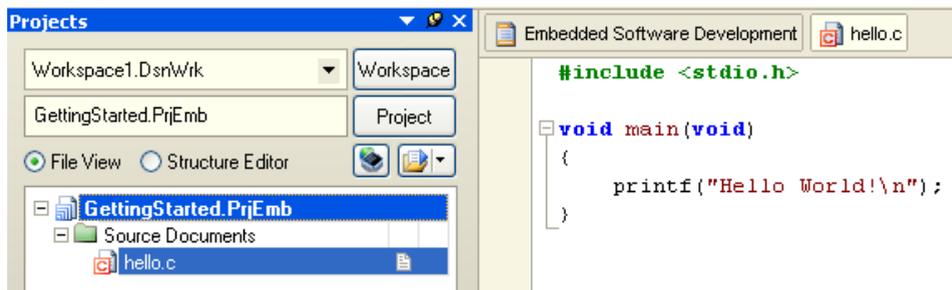
For a new assembly file select **Assembly File** instead of **C File** and for a new text file select **Text Document**.

2. Enter the source code required. For this tutorial enter the following code:

```
#include <stdio.h>
void main(void)
{
    printf("Hello World!\n");
}
```

3. Save the source file by selecting **File » Save As**. Navigate to where you want to store the source file on your hard disk, type the name `hello.c` in the File name field and click on **Save**.
4. Save the project by right-clicking on `GettingStarted.PrjEmb` in the **Projects** panel and select **Save Project**.

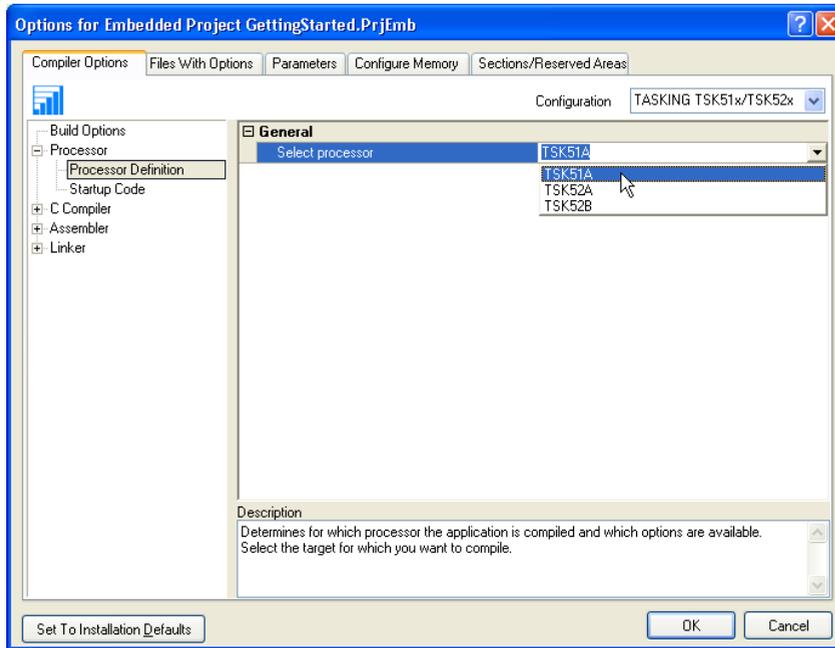
Your project now looks like:





## Getting Started with Embedded Software

2. In the **Compiler Options** tab, select the **Configuration** (for example, **TASKING TSK51x/TSK52x**).
3. In the left pane, expand the **Processor** entry and select **Processor Definition**.



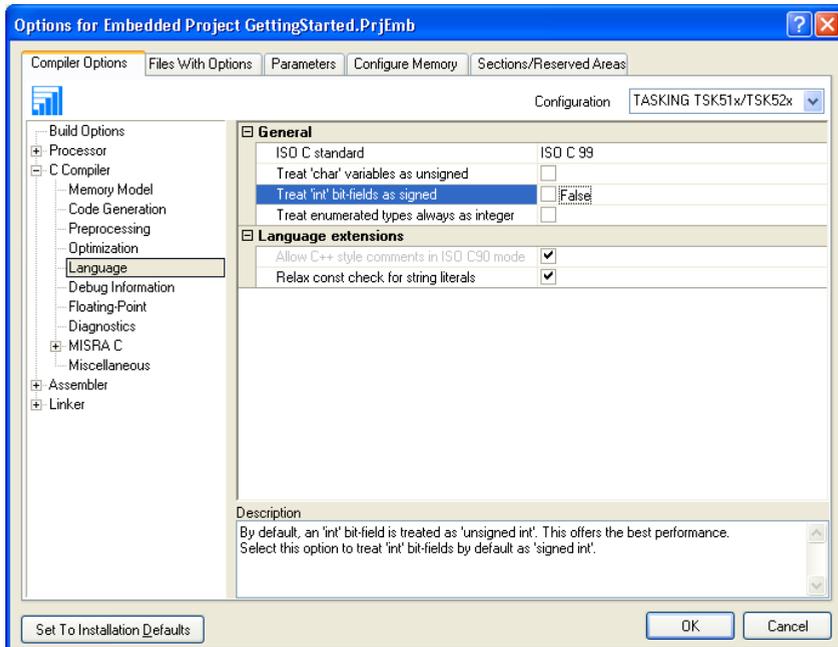
4. In the right pane, expand the **General** entry and set **Select processor** to the correct target processor (for example, **TSK51A**).
5. Click **OK** to accept the new target processor.

## Setting the tool options

You can set embedded options commonly for all files in the project and you can set file specific options.

### Setting project wide options

1. In the **Projects** panel, right-click on `GettingStarted.PrjEmb` and select **Project Options**. Alternatively, select **Project » Project Options** from the menu.  
The *Options for Embedded Project* dialog appears.



2. In the left pane, expand the **C Compiler** entry. This entry contains several pages where you can specify C compiler settings.
3. In the right pane, set the options to the values you want. Do this for all pages.
4. Repeat steps 2 and 3 for the other tools like assembler and linker.
5. Click **OK** to confirm the new settings.

Based on the embedded project options, Altium Designer creates a so-called *makefile* which it uses to build your embedded application.

On the **Miscellaneous** page of each tool entry, the **Command line options** field shows how your settings are translated to command line options.

## Setting options for an individual document

1. In the **Projects** panel, right-click on `hello.c` and select **Document Options**. Alternatively, select **Project » Document Options** from the menu. The *Options for Document* dialog appears.

Steps 2 to 5 are the same as the steps for setting project wide options. The **Files With Options** tab in the *Options for Embedded Project* dialog shows which files have deviating settings. If you right-click on a file in this tab, a menu provides you with functions to copy options quickly from and to other individual files.

## Building your Embedded Application

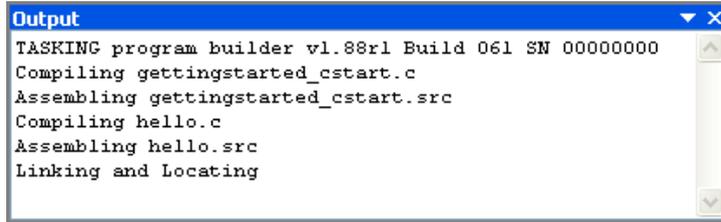
You are now ready to build your embedded application.

1. Select **Project » Compile Embedded Project GettingStarted.PrjEmb** or click on the  button.

## Getting Started with Embedded Software

The TASKING program builder compiles, assembles, links and locates the files in the embedded project that are out-of-date or that have been changed during a previous build. The resulting file is the absolute object file `GettingStarted.abs`.

2. You can view the results of the build in the Output panel (**Panels » System » Output**).



The screenshot shows a window titled "Output" with a blue header bar. The text inside the window is as follows:

```
TASKING program builder v1.88r1 Build 061 SN 00000000
Compiling gettingstarted_cstart.c
Assembling gettingstarted_cstart.src
Compiling hello.c
Assembling hello.src
Linking and Locating
```

## Compiling a single source file

If you want to compile a single source file:

1. Right-click on the file (`hello.c`) you want to compile and select **Compile Document hello.c**. Alternatively, you can open a file in the Text Editor and select **Project » Compile Document hello.c**.
2. Open the Messages panel to view any errors that may have occurred during compilation by selecting **View » Workspace Panels » System » Messages**, or selecting **System » Messages** from the **Panels** tab.
3. Correct any errors in your source files. Save your project files.

## Rebuilding your entire application

If you want to build your embedded application from scratch, regardless of their date/time stamp, you can perform a recompile:

1. Select **Project » Recompile Embedded Project GettingStarted.PrjEmb**.
2. The TASKING program builder compiles, assembles, links and locates all files in the embedded project unconditionally.

You can now debug the resulting absolute object file `GettingStarted.abs`.

## Debugging your Embedded Application

---

When you have built your embedded application, you can start debugging the resulting absolute object file with the simulator.

To start debugging, you have to execute one or more source lines:

- Select one of the source level or instruction level step options (**Debug » Step Into, Step Over**) to step through your source or select **Debug » Run** to run the simulation.

*A blue line indicates the current execution position.*

To view more information about items such as registers, locals, memory or breakpoints, open the various workspace panels:

- Select **View » Workspace Panels » Embedded » (a\_panel)**.

To end a debugging session:

- Select **Debug » Stop Debugging**.

## Setting breakpoints

You can set breakpoints when the embedded source file is opened. Small blue points indicate where you can set breakpoints:

- Click on the left margin next to the source line to toggle a breakpoint on and off.  
*A red crossed circle and red line mark the breakpoint.*

To change the breakpoint's properties:

- To change the breakpoint, right-click on the breakpoint and select **Breakpoint Properties...**

To disable or enable a breakpoint:

- On the breakpoint and select **Disable Breakpoint** (or **Enable Breakpoint** when it was disabled)  
*A disabled breakpoint is marked with green color.*

The breakpoint panel gives an overview of all (disabled) breakpoint and their properties:

- Select **View » Workspace Panels » Embedded » Breakpoints**.

## Evaluating expressions

You can examine the value of expressions in the **Evaluate** panel.

1. Open the **Evaluate** panel by selecting **View » Workspace Panels » Embedded » Evaluate**.
2. In the edit field enter the expression you want to evaluate and click **Evaluate**.

The expression and its value appear below in the **Evaluate** panel.

To watch an expression continuously, you can set a *watch*:

3. Select **Add Watch**

The new expression and its value appear in the **Watches** panel.

*Alternatively:* Select **Debug » Add Watch**; enter an expression and click **OK**.

The way an expression is evaluated, depends strongly on the amount of debug information in the object file. Also the optimization level influences the ease of debugging.

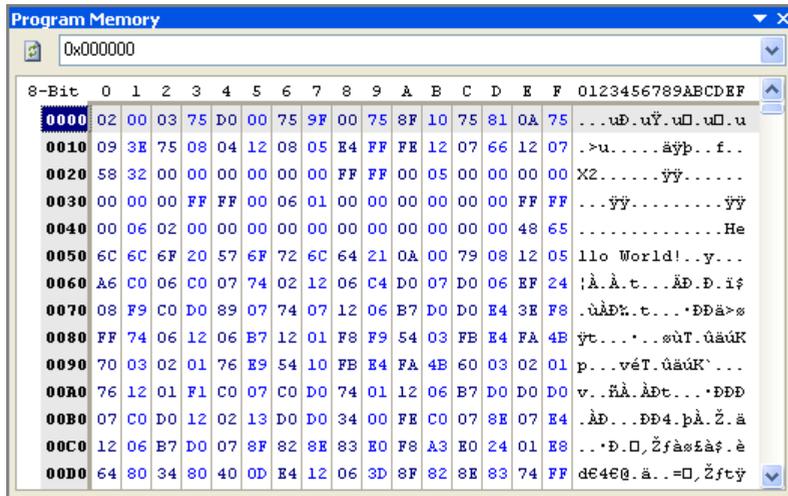
## Viewing memory

It is possible to view the contents of the memory. You can open several memory windows. Remember, you must be in debugging mode. The type of memory windows you can open, depends on the selected target processor.

To open for example a program memory window:

1. Select **View » Workspace Panels » Embedded » Program Memory**.  
*The Program Memory window opens showing the contents of the code memory.*
2. In the edit field you can specify the address you want to start viewing from.

## Getting Started with Embedded Software



## Revision History

Date	Version No.	Revision
20-Jan-2004	1.0	New product release
22-Apr-2005	1.1	Changes/updates according to SP3 release.
08-Jun-2005	1.2	Changes/updates according to SP4 release.

Software, hardware, documentation and related materials:

Copyright © 2005 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment. Altium, Altium Designer, CAMtastic, Design Explorer, DXP, LiveDesign, NanoBoard, NanoTalk, Nexar, nVisage, CircuitStudio, P-CAD, Protel, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.