



Getting Started with Embedded Software

Summary

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

Tutorial

TU0122 (v2.0) April 21, 2008

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. An embedded software project is normally used as a subproject for an FPGA project and once they are built, they are downloaded and executed inside an FPGA device.

The example used in this tutorial is a Hello World program in C. Other examples are supplied in the `\Examples\NanoBoardCommon\Processors Examples` folder relative to the installation path.

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, TSK3000, MicroBlaze, Nios II and ARM. Figure 1 shows all components of the TASKING toolset with their input and output files.

The C compiler, assembler, linker and debugger are target dependent, whereas the librarian is target independent. The **bold** names in the following figure are the executable names of the tools. Substitute **target** with one of the supported target names, for example, **cppc** is the PowerPC C compiler, **c3000** is the TSK3000 C compiler, **as165x** is the TSK165x assembler, etc.

Getting Started with Embedded Software

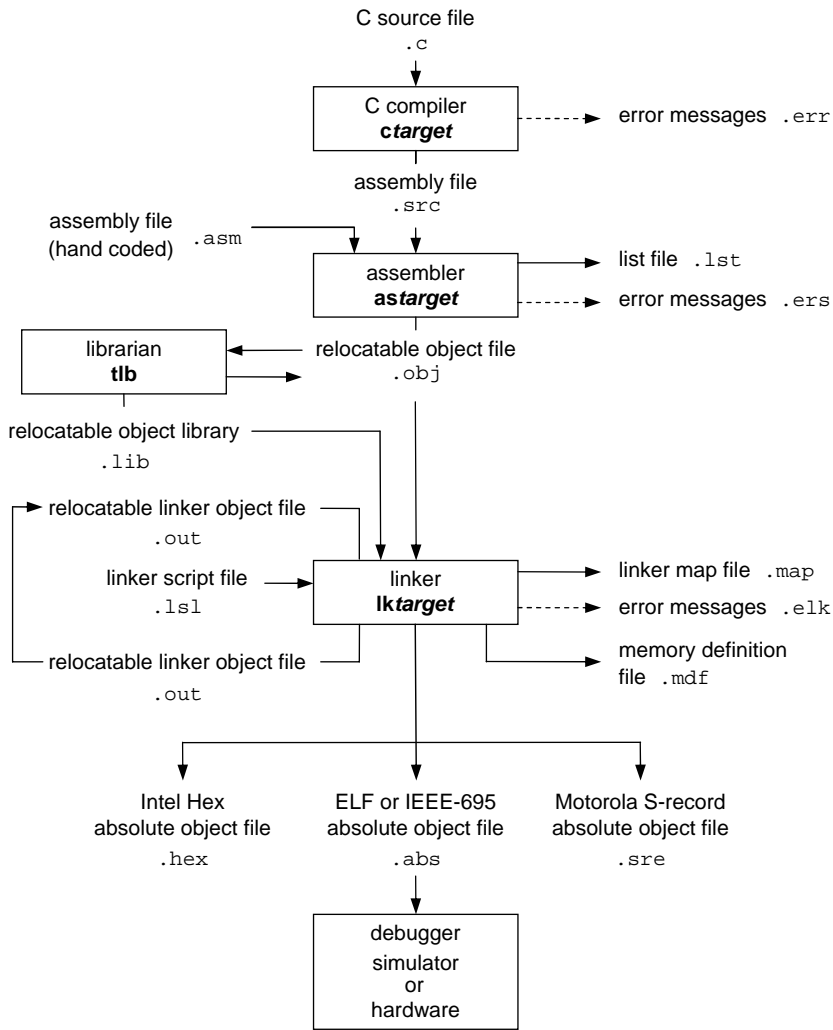


Figure 1: Toolset overview

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

Extension	Description
Source files	
.c	C source file, input for the C compiler
.asm	Assembler source file, hand coded
.lsl	Linker script file
Generated source files	
.src	Assembler source file, generated by the C compiler, does not contain macros
Object files	
.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
List files	
.lst	Assembler list file
.map	Linker map file
.mcr	MISRA-C report file
.mdf	Memory definition file
Error list files	
.err	Compiler error messages file
.ers	Assembler error messages file
.elk	Linker error messages file

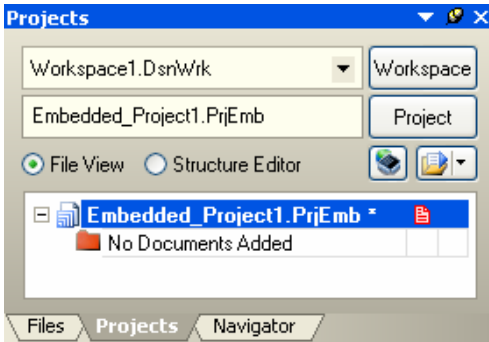
Table 1: File extensions

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.
2. The **Projects** panel displays a new project file, `Embedded_Project1.PrjEmb`.



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

```
#include <stdio.h>

void printloop(void)
{
    int loop;

    for (loop=0; loop<10; loop++)
    {
        printf("%i\n", loop);
    }
}

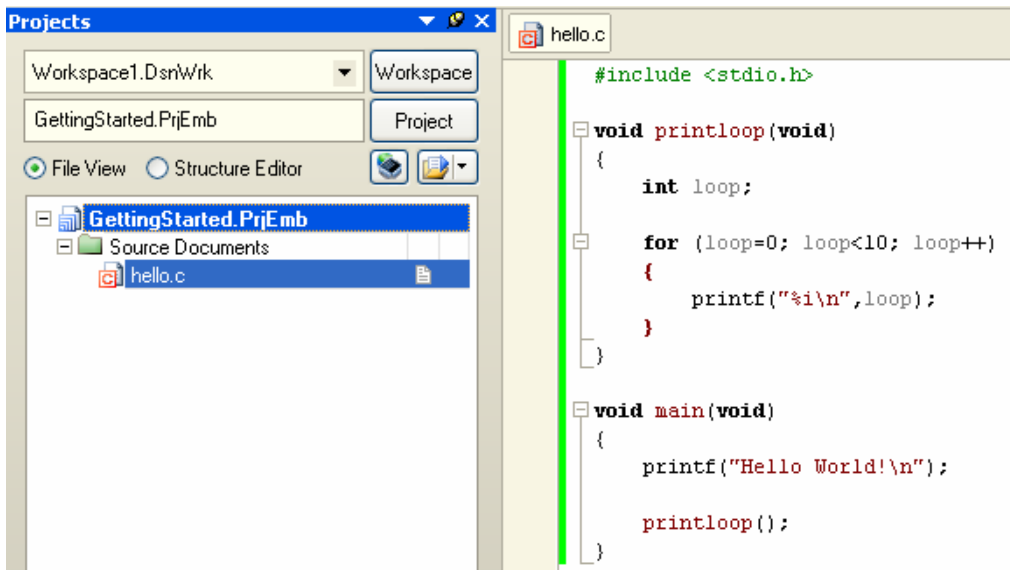
void main(void)
{
    printf("Hello World!\n");

    printloop();
}
```

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

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:



Adding existing source files to the project

If you want to add an existing source file to your project, proceed as follows:

1. In the **Projects** panel, right-click on the project name and select **Add Existing to Project**. The *Choose Documents to Add to Project* dialog appears.
2. Navigate to the file you want to add to your project and click **Open**.
3. The source file is added to the project and listed in the **Projects** panel. Double-click on the filename to view or edit this file in the Text Editor.
4. Save the project (right-click on the project name in the **Projects** panel and select **Save Project**).

Setting the Embedded Project Options

An embedded project in Altium Designer has a set of embedded options associated with it. After you have added files to your project, and have written your application (`hello.c` in our example), the next steps in the process of building your embedded application are:

- selecting a device (resulting in an associated toolset)
- specifying the options of the tools in the toolset, such as the C compiler, assembler and linker options. (Different toolsets may have different sets of options.)

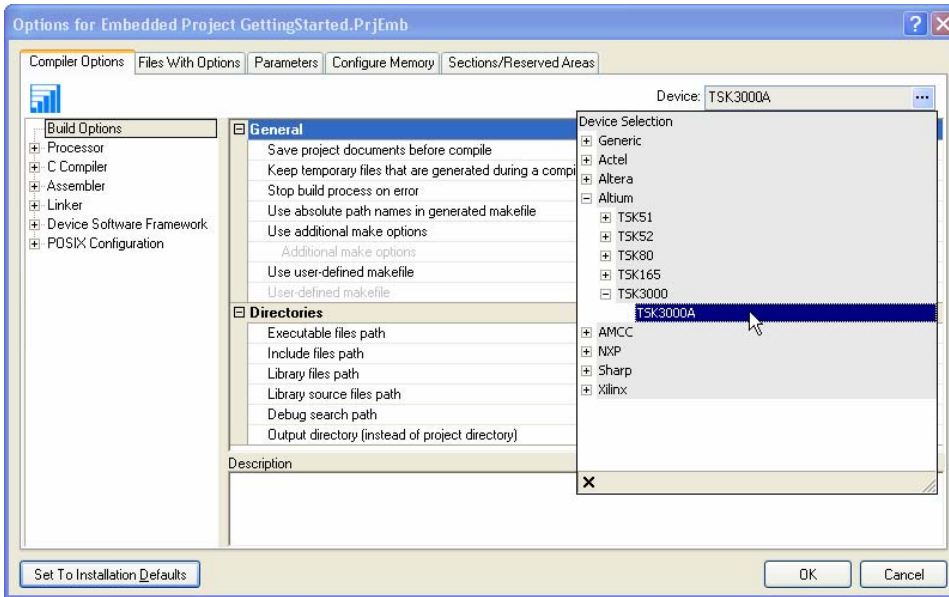
Selecting a device

For an embedded project, you must specify the device for which you want to build your embedded project first:

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.

Getting Started with Embedded Software



- In the **Compiler Options** tab select the **Device**. You can make a selection based on manufacturer, or you can select a generic device. If you select a device from a manufacturer, the correct processor type is selected automatically. If you select a generic device, you have to specify the target processor type manually.

To specify the target processor type manually (only for a Generic device):

- In the left pane, expand the **Processor** entry and select **Processor Definition**.
- In the right pane, expand the **General** entry and set **Select processor** to the correct target processor.
- Click **OK** to accept the new device.

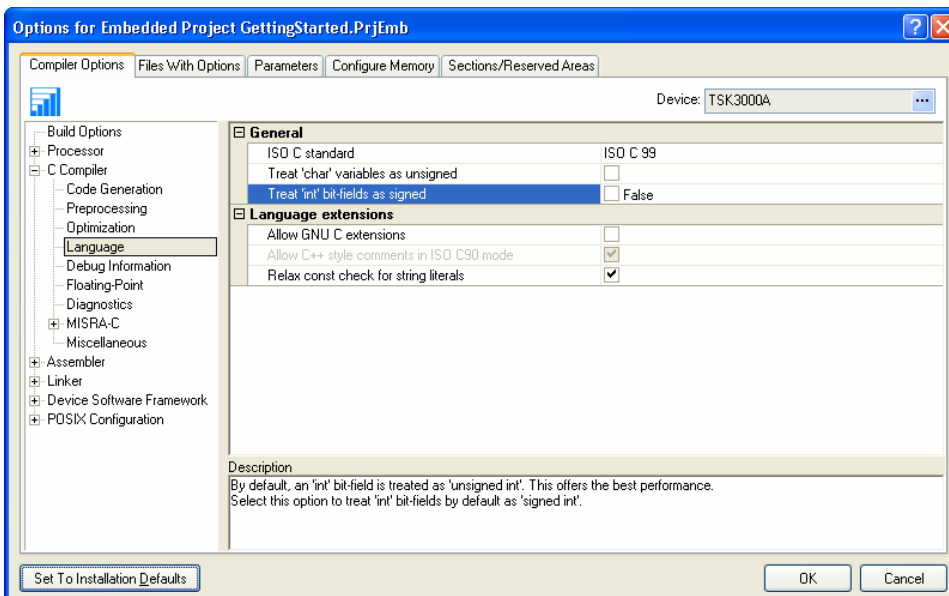
Setting the tool options

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

Setting project wide options

- 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.



- In the left pane, expand the **C Compiler** entry. This entry contains several pages where you can specify C compiler settings.
- In the right pane, set the options to the values you want. Do this for all pages.

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

- Repeat steps 2 and 3 for the other tools like assembler and linker.
- 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.


Setting options for an individual document

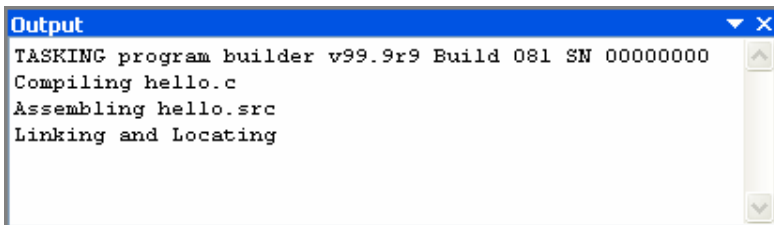
- In the **Projects** panel, right-click on `hello.c` and select **Document Options**. Alternatively, select **Project » Document Options** from the menu. The *Options for Document* dialog appears.

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.

- Select **Project » Compile Embedded Project GettingStarted.PrjEmb** or click on the  button.
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`.
- You can view the results of the build in the **Output** panel (**View » Workspace Panels » System » Output**).



Compiling a single source file

If you want to compile a single source file:

- 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**.
- 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.
- 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:

- Select **Project » Recompile Embedded Project GettingStarted.PrjEmb**.
- 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

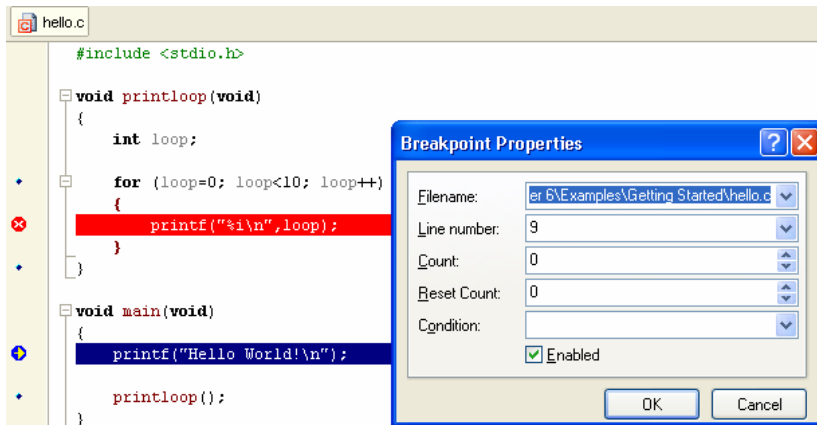
You can set breakpoints when the embedded source file is opened. Small blue points indicate where you can set breakpoints:

- 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:

- Right-click 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 and watching 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. Click on the **Evaluate** button every time the variable in the code is modified.

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. The values in the **Watches** panel are updated continuously when the code is executing.

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

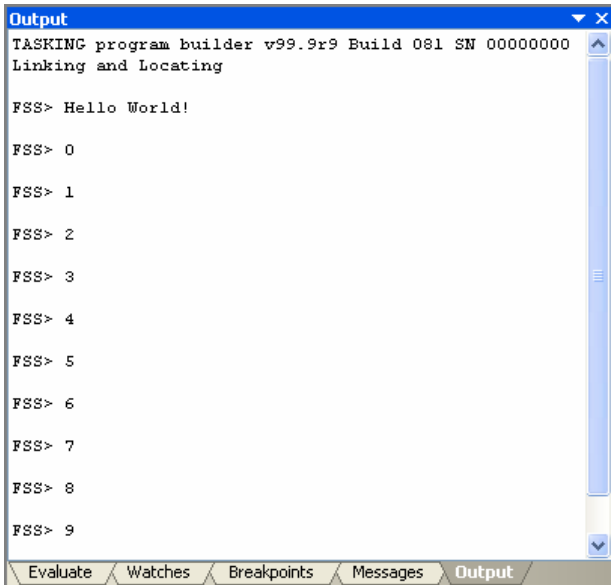
The way an expression is evaluated, depends strongly on the amount of debug information in the object file. Also the optimization level influences the ease of debugging.

Viewing output

It is possible to view the output generated by your embedded application. Remember, you must be in debugging mode. To open the **Output** panel:

- Select **View » Workspace Panels » System » Output**.

The Output panel opens showing the output of the embedded application.



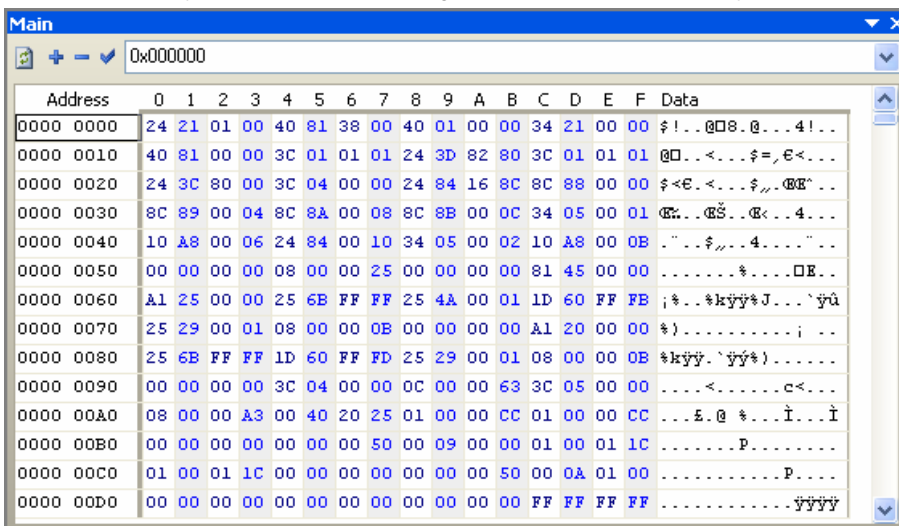
Viewing memory

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 the Main memory window:

1. Select **View » Workspace Panels » Embedded » Main**.

The Main memory window opens showing the contents of the memory.



2. In the edit field you can specify the address you want to start viewing from.

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.
01-Nov-2005	1.3	Changes/updates according to Altium Designer 6 release.
09-Nov-2006	1.4	Changes/updates according to Altium Designer 6.6 release and information on Output panel expanded. Code example expanded to include a simple loop statement as well as the hello world statement.
12-Mar-2007	1.5	Changes/updates according to Altium Designer 6.7 release.
21-Apr-2008	2.0	Updates for Altium Designer Summer 08 release.

Software, hardware, documentation and related materials:

Copyright © 2008 Altium Limited. All Rights Reserved.

The material provided with this notice is subject to various forms of national and international intellectual property protection, including but not limited to copyright protection. You have been granted a non-exclusive license to use such material for the purposes stated in the end-user license agreement governing its use. In no event shall you reverse engineer, decompile, duplicate, distribute, create derivative works from or in any way exploit the material licensed to you except as expressly permitted by the governing agreement. Failure to abide by such restrictions may result in severe civil and criminal penalties, including but not limited to fines and imprisonment. Provided, however, that you are permitted to make one archival copy of said materials for back up purposes only, which archival copy may be accessed and used only in the event that the original copy of the materials is inoperable. Altium, Altium Designer, Board Insight, DXP, Innovation Station, LiveDesign, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, 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. v8.0 31/3/08