

Converting CodeWarrior HC(S)08 V3.1 Projects to CodeWarrior HC(S)08 V5.0

Essentially you can open, build, and debug a project previously created with CodeWarrior for HC08 V3.1 with CodeWarrior for HC08 V5.0 without any problem.

- CodeWarrior for HC08 V5.0 has been updated with a new feature allowing to switch easily between connections and derivatives. This Technical Note explains how to [extend a V3.1 project with the new functionality of the V5.0 tool chain](#).
- Connections have been updated for CodeWarrior for HC08 V5.0 (Some have been suppressed):
 - [Using new Connections available with the new release of the tool](#).
 - [Behavior of the debugger when the connection to the P&E Hardware connections fails](#)
 - [How to deal with obsolete connections](#).
- Some behaviors have changed between CodeWarrior for HC08 V3.1 and CodeWarrior for HC08 V5.0:
 - [Debugging session start](#).
 - [How to get a similar project layout as in CodeWarrior for HC08 V3.1 with CodeWarrior for HC08 V5.0](#).
 - [Working with build targets in CodeWarrior for HC08 V5.0 projects](#).

Terms and Abbreviations

The following terms and acronyms are used in the document.

MCU:	An HC08 or an HCS08 derivative (e.g., an HCS908GB60)
Connection;	The probe used to connect to the processor being debugged. Sometimes also called the target connection device (e.g., Full chip Simulator, P&E Multilink/Cyclone Pro)
V5.0:	Refers to CodeWarrior Development Studio for Freescale HC08 5.0
V3.1:	Refers to CodeWarrior Development Studio for Motorola HC08 3.1
FCS:	Full-Chip Simulation
ICD:	In-Circuit Debugging

Extend a V3.1 project with the new functionality of the V5.0 tool chain

Restriction:


- You will not be able to apply the changes described below to a Processor Expert project created with V3.1.
- You will not be able to apply the changes described below to an OSEK project.
- Also if you are not using a standard PRM file (if you have added some commands, using some user defined sections or memory area), extensions to the PRM file will get lost when you attempt to change to another derivative.

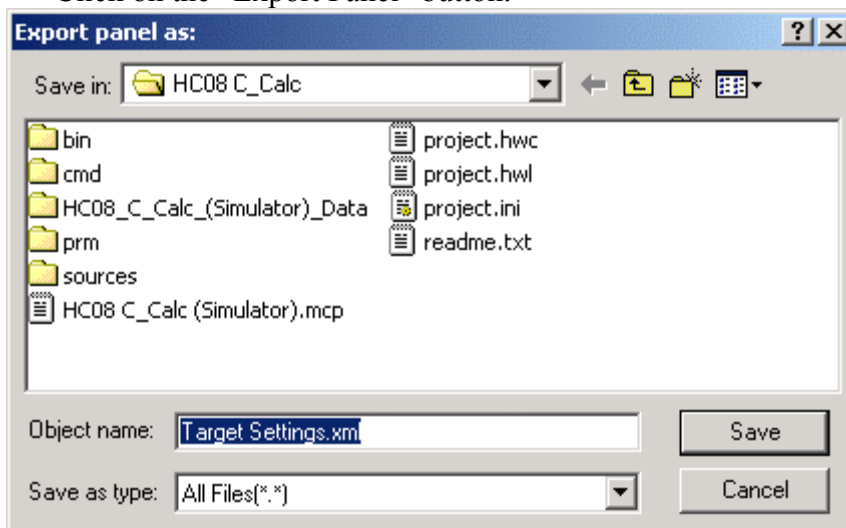
Activating Change MCU and Connection

Projects created by the Wizard in V5.0 include support to easily switch to a new MCU or a new connection. You will find instructions below on how to add this functionality to your existing project created using V3.1 or earlier.

We recommend that you create a new project with the latest Wizard and then restore your project settings in the new project.

1. *Export Target Settings to XML files*

- Open the original project with V3.1 IDE
- Select the Build Target Setting dialog box (Press ALT + F7 or click the Target setting icon  in the project window's toolbar).
- Export settings from the “Target Settings” panel to “Target Settings.xml”.
 - Click on “Target Settings” to see the “Target Settings” preference panel.
 - Click on the “Export Panel” button.



- Save Panel Settings as “Target Settings.xml”.
- Export settings for the “Assembler for HC08” panel to “Assembler for HC08.xml”.
- Export settings for the “Compiler for HC08” panel to “Compiler for HC08.xml”.
- Export settings for the “Importer for HC08” panel to “Importer for HC08.xml”.
- Export settings for the “Linker for HC08” panel to ”Linker for HC08.xml”.

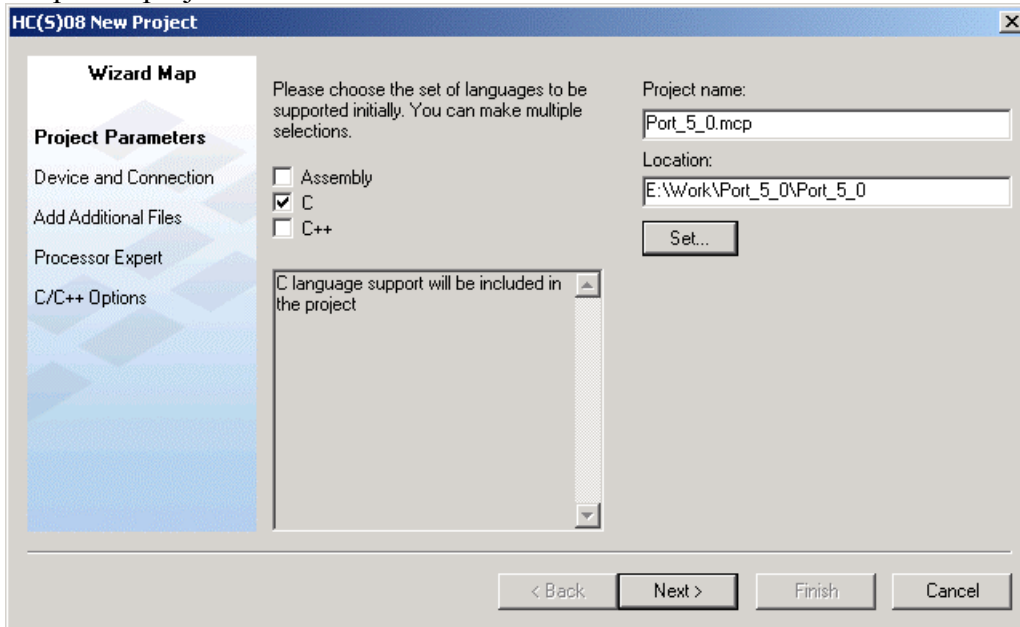
- If you are using OSEK, export settings for the “OSEK Sysgen” panel to “OSEK Sysgen.xml”.
- In case you are using PC-lint, export settings for the “PCLint Main Settings” panel to “PCLint Main Settings.xml” and “PCLint options” to “PCLint Options.xml”.
- If you are using non-standard file extensions, export settings for the “File Mappings” panel to “File Mappings.xml”.
- In case you are using Source Trees, export settings for the “Source Trees” panel to “Source Trees.xml”.
- If you are using custom Keywords, export settings for the “Custom Keywords” panel to “Custom Keywords.xml”.

2. Create a New Project using the V5.0 Wizard

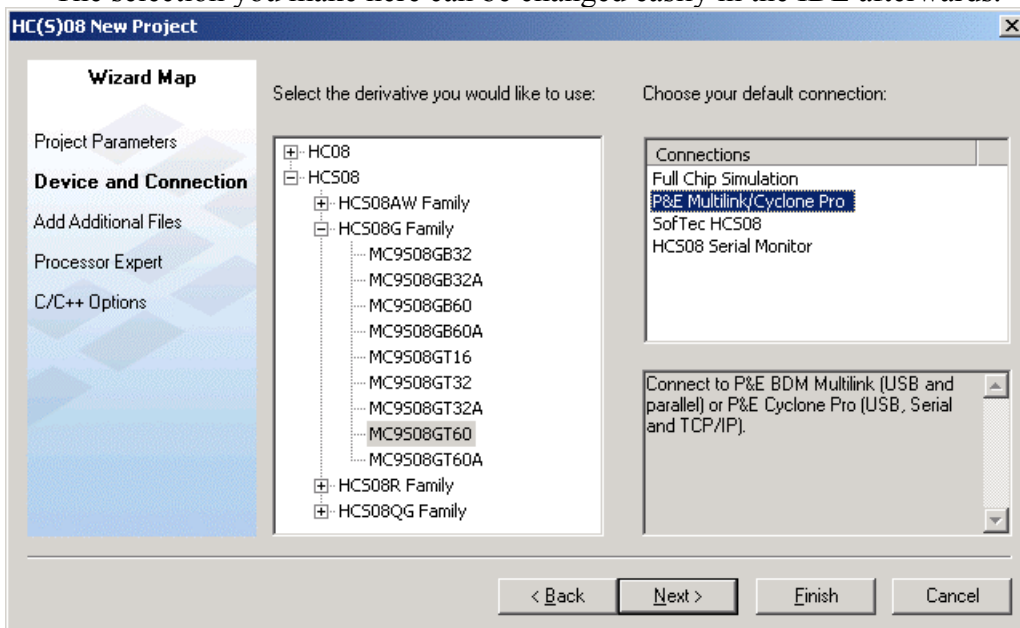
- Start the V5.0 IDE
- Click on the “Create New Project” button in the Startup dialog box.



- In the “Project Parameters” panel
 - Check the boxes in front of any programming languages used in the project.
 - Use the “Set ...” button to browse for the location where you want to store your V5.0 project. For backward compatibility, we recommend that you create a new directory for the ported project.

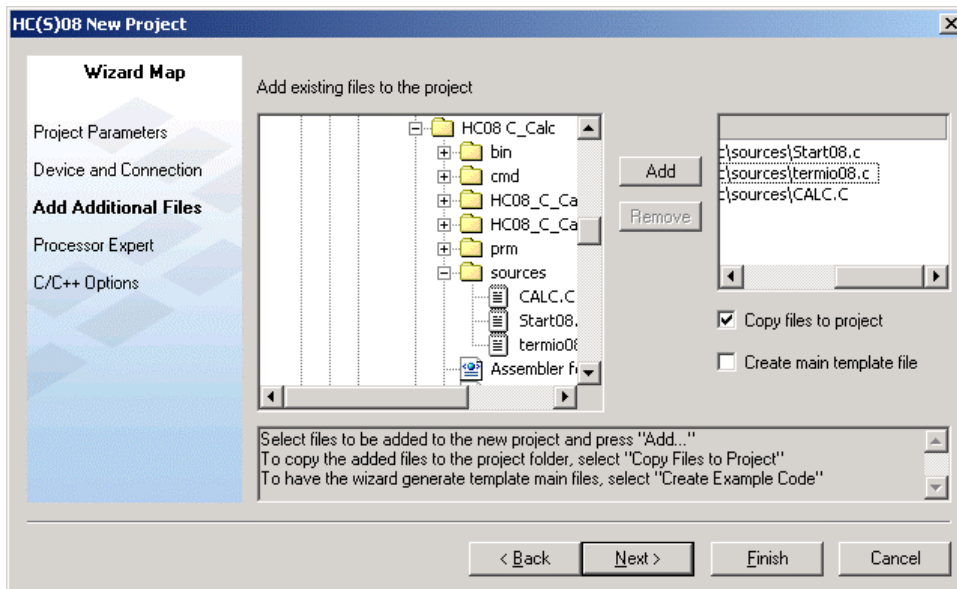


- Click the “Next >” button
- In the “Device and Connection” panel, select the derivative and the default connection you wish to use in the project.
 - The selection you make here can be changed easily in the IDE afterwards.

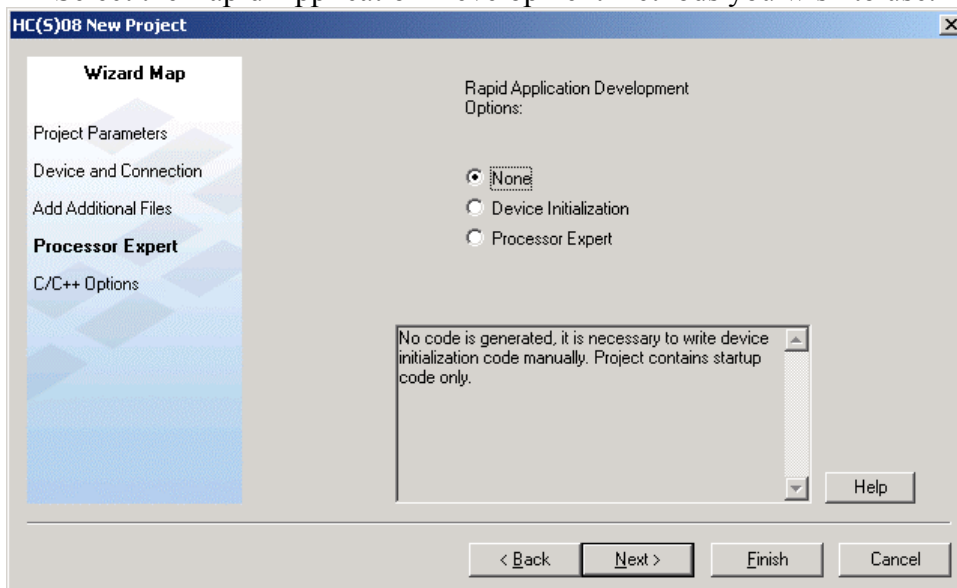


- Click on the “Next >” button.
- In the “Add Additional Files” panel:

- Browse for the original project's source files (or the source file directory).
- Select each of the source files in the original project and press “Add”.
- From here you can decide if you want to duplicate the application source files in the new project directory or if you want to use the original source files in the new project.
- We would recommend duplicating the source file. So make sure the “Copy files to Project” check box is checked.

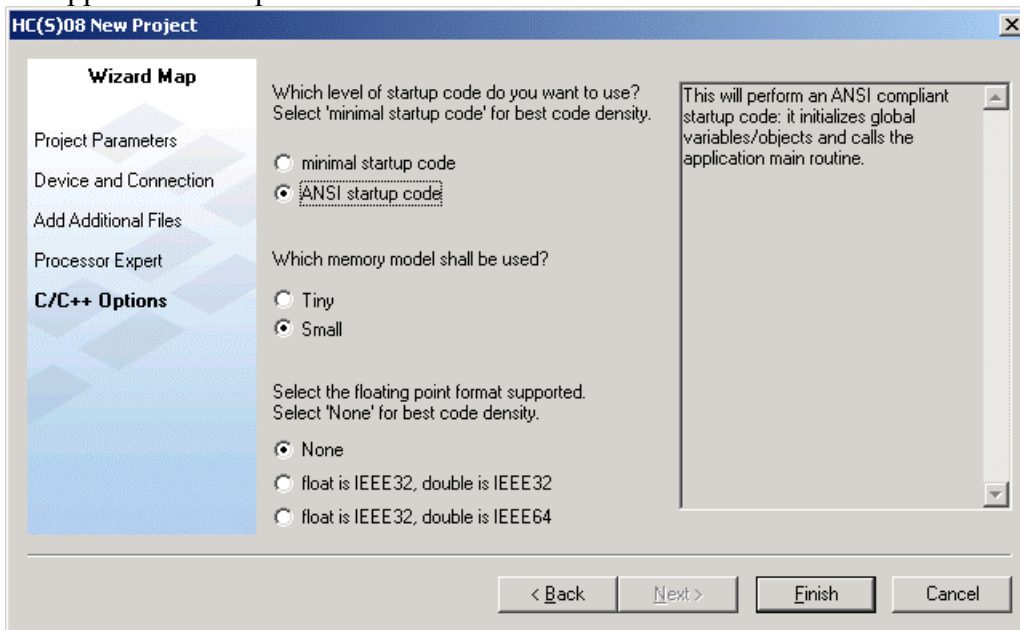


- Uncheck the box “Create main template file” as your current application already include a main function.
- Click on the “Next >” button.
- The “Processor Expert” panel might not be available for all derivatives. In this panel:
 - Select the Rapid Application Development methods you wish to use.



- If you were not using Processor Expert before, select “None”.

- If you were using Processor Expert before keep your existing project. The porting instructions in this section do not apply (See Restrictions above).
- Click on the “Next >” button.
- In the “C/C++ Options” panel:
 - Select the level of startup code, memory model, and floating-point format according to the application's requirements.




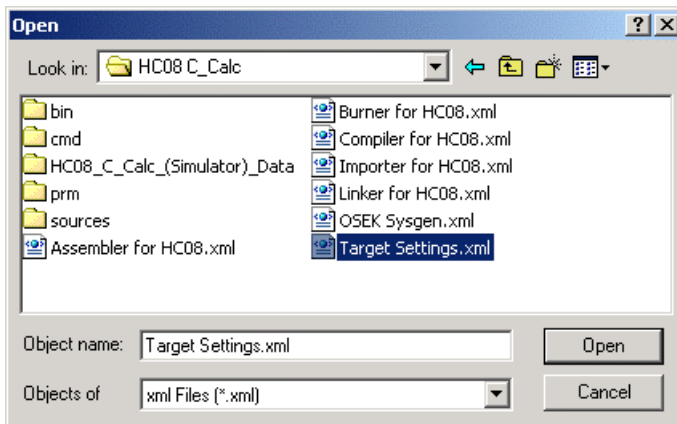
- Click on the “Next >” button.
- Finally in order to enable the “Change MCU/Connection” feature, in all source & header file within the application change the include for the I/O register definition file to `#include "derivative.h"`.
- Do not change the include statement in the MC*.c file from the Libs Group.

A new project has now been created for V5.0, in which the “Change MCU/Connection” from the Project menu has been enabled.

3. Import Target Settings from XML Files

Now you need to apply your original project settings to the new project.

- Select the Build Target Setting dialog box (Press ALT + F7 or click the Target setting icon  in the project window toolbar).
- Import settings for the “Target Settings” panel from “Target Settings.xml”.
 - Click on “Target Settings” to see the “Target Settings” panel.
 - Click on the “Import Panel” button.



- Open “Target Settings.xml”.
- Import settings for the “Assembler for HC08” panel from “Assembler for HC08.xml”.
- Import settings for the “Compiler for HC08” panel from “Compiler for HC08.xml”.
- Import settings for the “Importer for HC08” panel from “Importer for HC08.xml”.
- Import settings for the “Linker for HC08” panel from “Linker for HC08.xml”.
- If you are using OSEK, import settings for the “OSEK Sysgen” panel from “OSEK Sysgen.xml”.
- In case you are using PC-lint, import settings for the “PCLint Main Settings” panel from “PCLint Main Settings.xml” and “PCLint options” from “PCLint Options.xml”.
- If you are using non-standard file extensions, import settings for panel “File Mappings” from “File Mappings.xml”.
- In case you are using Source Trees, import settings for the “Source Trees” panel from “Source Trees.xml”.
- If you are using custom Keywords, import settings for the “Custom Keywords” panel from “Custom Keywords.xml”.

Now you have imported all your original project's settings for configuring the new project.

4. *Debugger Settings.*

Debugger settings are not easily portable from V3.1 to V5.0. All debugger-specific settings from the original V3.1 project must be applied manually to the new V5.0 project. The following points should be taken into consideration:

- **Debugger Command Line:**
Do not add any options on the debugger command line in a project created with CodeWarrior V5.0 . The command line will be changed completely when you switch to another connection.
- **Debugger Command Files:**
If you were using some commands in the debugger command files, you have to apply them again to the new command files.
You can copy the contents of the original command files to the new files.
- **Connection settings:**
If you were using specific connection settings in the original project, you must apply them again using the “Communication” menu entry in the Target menu (Multilink/CyclonePro, FCS, Softec...).

Connections

Using new connections

The PEDebug Connection is now being deprecated. P&E Connections are now covered by a set of new connections.

The GDI connection has also been deprecated. New connections with meaningful names have been defined in V5.0.

1. *HC08*

- **Full Chip Simulation:**

P&E HC08 Full Chip Simulation with simulation of all on-chip peripherals. Matches the V3.1 PEDebug -> “Mode” -> “Full Chip Simulation” configuration.

- **Mon08 Interface:**

Mon08 Interface (through serial port, for P&E classes I to IV). Connect to Motorola ICS boards or direct serial connection (MON08). Matches the V3.1 PEDebug -> “Mode” -> “In Circuit Debug/Programming” configuration.

- **P&E Multilink/Cyclone Pro:**

Connect to P&E MON08 Multilink (USB and parallel-port) or P&E Cyclone Pro (USB, Serial, and TCP/IP). Matches the V3.1 PEDebug -> “Mode” -> “In Circuit Debug/Programming” configuration.

- **SofTec HC08:**

Connect to any of the USB-based SofTec Microsystems tools for HC08 (inDART-HC08, etc.).

2. *HCS08*

- **Full Chip Simulation:**

P&E HCS08 Full Chip Simulation with simulation of all on-chip peripherals. Matches the V3.1 PEDebug -> “Mode” -> “Full Chip Simulation” configuration

- **P&E Multilink/Cyclone Pro:**

Connect to P&E BDM Multilink (USB and parallel-port) or P&E Cyclone Pro (USB, Serial, and TCP/IP). Matches the V3.1 PEDebug -> “Mode” -> “In Circuit Debug/Programming” configuration.

- **SofTec HCS08:**

Connect to any of the USB-based SofTec Microsystems tools for HCS08 (inDART-HC08, etc.).

- **HCS08 Serial Monitor:**

Connect to a board through the Freescale HCS08 Serial Monitor.


The first time you try to debug a project created with V3.1 on V5.0, the debugger will automatically switch to those connections according to the settings in the old debugger project files.

Connection to the P&E Hardware connections fails

In V3.1, PEDebug target interface was seamlessly switching between ICD and FCS.

This is no longer the case with V5.0. In case the user cannot connect, the debugger stays unconnected and waits for the user to set another connection or to refine his connection settings.

In order to switch from FCS “Full Chip Simulation” to “P&E Multilink/Cyclone Pro” connection (or vice versa), one should:

- Close the Debugger
- In the IDE, select “Project” -> “Change MCU/Connection”
- Click the Debug button () in the tool bar.

Dealing with Obsolete Connections

Some other connections have been suppressed from V3.1 to V5.0:

- **Lauterbach, Hitex, and Mon08:** Connections have been completely removed from the layout. We recommend that you use the Lauterbach Trace32 Debugger if you are debugging through a Lauterbach emulator or BDM. You should use the Hitex HITOP Debugger if you are debugging through a Hitex emulator or BDM. We recommend that you use the “Mon08 Interface” connection if you are debugging through the Mon08 on-chip monitor.
- **The GDI Connection** is now hidden in the Set Connection dialog box. There are now separate connections for Softec and Serial Monitor.
- **HC(S)08, Freescale Simulator Connection** (i.e., HIWARE ISS) is now hidden in the Set Connection dialog box.
- **ICS Mon08 Interface:** In-Circuit Simulation, using Mon08 Interface through serial port, for P&E class I to IV. This connection is only available in the Debugger Set Connection dialog.
- **ICS P&E Multilink/Cyclone Pro:** In-Circuit Simulation, using P&E MON08 Multilink (USB and parallel) or P&E Cyclone Pro (USB, Serial and TCP/IP). This connection is only available in the Debugger Set Connection dialog.

New Behavior

The current section describes the change in the behavior of the tools between CodeWarrior HC08 V3.1 and V5.0.

Debugging Session Start

For new project created with V5.0, the default behavior when you start a debugging session has changed somewhat.

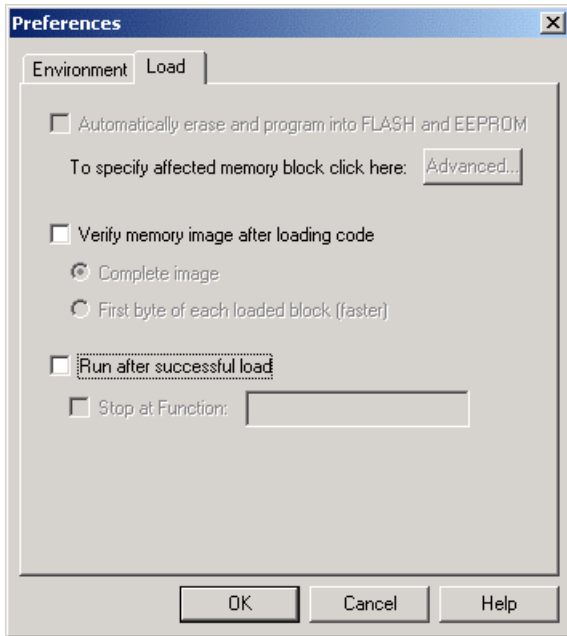
In V3.1, after the code has been downloaded, the Debugger showed the source code for the main function in the Source Window. The PC was still pointing to the application entry point.

For projects newly created with V5.0, after the code has been downloaded, the Debugger runs the application up to the main function.

If you wish to return to the V3.1 behavior in your V5.0 project:

- Edit the postload command file (called <Connection>_postload.cmd).
- Enter the following command there:
`FindProc main`
- Save the file.
- Start the Debugger.

- Select “File” -> “Configuration”. The “Preference” dialog box is opened.
- Switch to the “Load” tab.



- Uncheck the box “Run after successful load”.
- Click on “OK”.

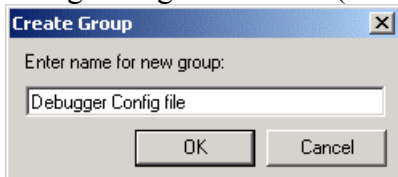
From now on the Debugger will behave exactly like the V3.1 debugger.

Getting same layout as V3.1

A programmer accustomed to the V3.1 project layout might be confused because the V5.0 layout does not include debugger command files and debugger configuration files in the project.

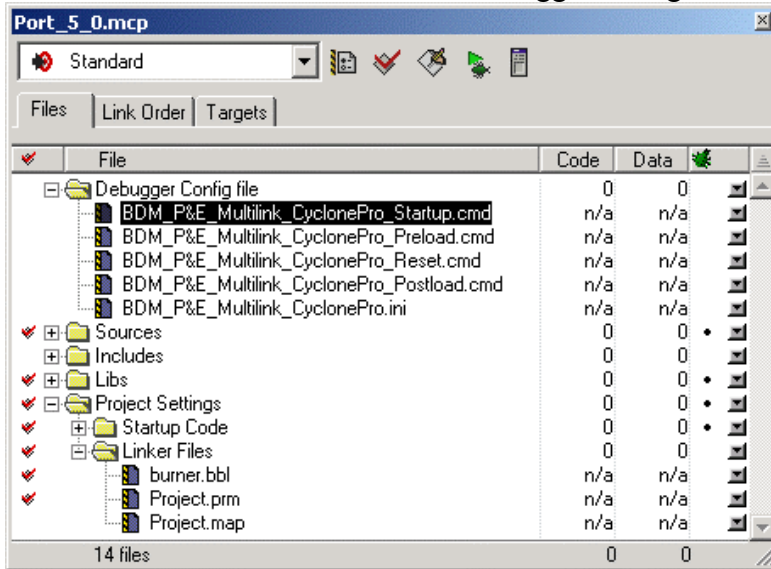
In order to add debugger configuration and command files to the project:

- Click with the right mouse button in the .mcp window
- In the pop up menu, select “Add Group”.
- In the “Create Group” dialog box, specify a logical name for the folder that will contain the debug configuration files (for example, “Debugger Config file”).



- Click OK. The new folder is displayed in the mcp window.
- Click with the right mouse button on the newly created Group (or folder).
- In the pop up menu, select “Add Files ...”.
- In the “Select Files to Add ...” dialog box, browse for the {Project}\cmd folder.
- Select all the .cmd files located there and click on the Open button. The files are added to the project’s “Debugger Config file” Group.
- Click with the right mouse button on the newly created Group (or folder).
- In the pop up menu, select “Add Files ...”.

- In the “Select Files to Add ...” dialog box, browse for the {Project}*.ini file.
- Select the file and click on OK. The debugger configuration file is added to the project.



Be careful:

When you switch to a new connection, the debugger will use another set of command and configuration files. So after changing the connection you might have to add new commands and configuration files to your project. The naming convention for the debugger configuration and command files is as follows. (A prefix is added according to the connection selected.):

1. HCS08

Connection	Prefix
Full Chip Simulator	HCS08_Full_Chip_Simulator
P&E Multilink/Cyclone Pro	BDM_P&E_Multilink_CyclonePro
Softec HCS08	SofTec_HCS08
HCS08 Serial Monitor	HCS08_Serial_Monitor

2. HC08

Connection	Prefix
Full Chip Simulator	HC08_Full_Chip_Simulator
Mon08	MON08_Interface
P&E Multilink/Cyclone Pro	MON08_P&E_Multilink_CyclonePro
Softec HC08	SofTec_HC08
MMDS/MMEVS Emulator	MMDS_MMEVS
FSICE Emulator	FSICE_Emulator

Using Build Targets

1. Connection Dependent Build Targets

The V3.1 Wizard allowed you to select several connections (called Target Interfaces in V3.1) at project creation. The generated project then included a build target for each selected connection.

Because the V5.0 Wizard allows you to easily switch to a new connection, you do not need these multiple targets any longer after you have enabled the “Change MCU/Connection” functionality in your project. (See section “[Activating Change MCU and Connection](#)” above”.)

2. MCU Dependent Build Targets

The V5.0 Wizard provides you with an easy means to switch to another MCU. So MCU dependent build targets might not be necessary anymore after you have enabled the “Change MCU/Connection” functionality in your project. (See section “[Activating Change MCU and Connection](#)” above”.)

You might have to keep your MCU dependent build targets when you have conditional code, depending on the MCU in the build target source files.

Example

You are building your build target with `-DMR32` and within your build target source or include file you have conditional blocks like:

```
#ifdef MR32
...
#else
...
#endif
```

In this case we recommend that you keep your MCU dependent build target. Changing the MCU using the menu entry will not adjust the macro definition on the compiler command line and your application might not run properly after you change the MCU.

3. Dependent Build Targets and Sub-projects

If your projects include subprojects or dependent build targets, you will have to recreate the build targets and the dependencies in the new project.

The steps described in “[Activating Change MCU and Connection](#)” should be followed for the main project. Then the other build targets must be re-created manually.

If you intend to change the MCU for the main target, you have to change the MCU also in each primary target and subproject. This has to be done manually.

If you intend to change the connection only, you just need to do that on the main application. Primary build targets and subprojects normally only provide an object library, which is independent from any hardware connection.

Note 1:

The “Build Target” combo box has been removed from the project window toolbar. If you have a multiple build target project and you want to change the current build target you can

- Either select menu entry “Project” -> “Set Default Target” and then select the build target
- Or go to the Target Tab and select the build target there.

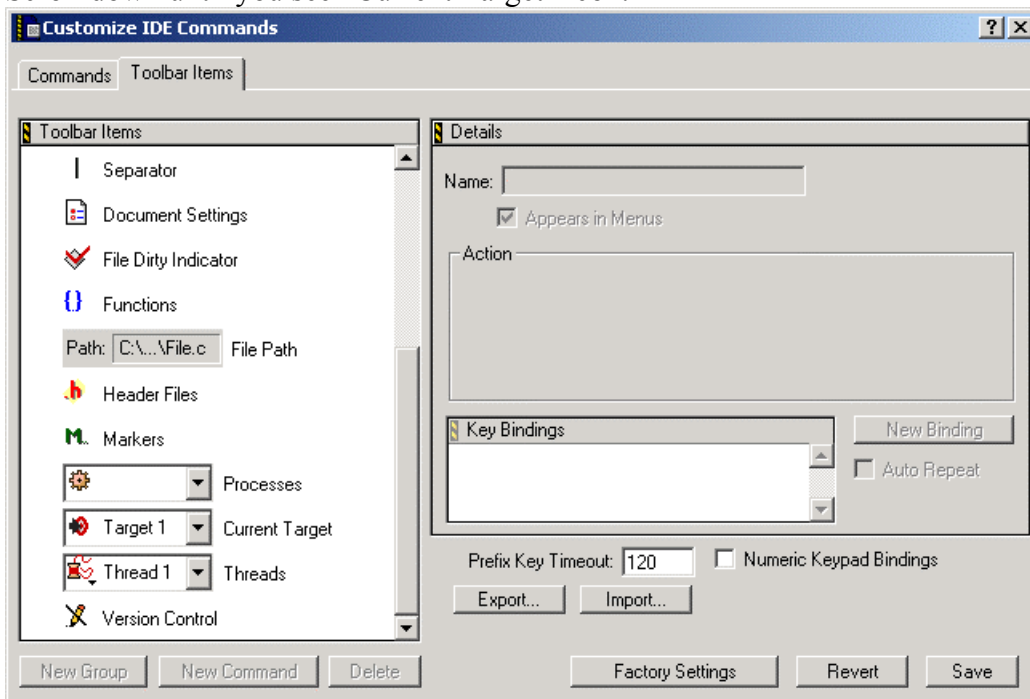
Note 2:

Basically if you are using Build targets for anything else than Connection Dependant Build Target or MCU Dependant Build target, we would recommend you to keep working with a V3.1 project.

4. Adding Build Target Combo Box to the Project Window

Would you wish to work with Build Targets in a project created with CodeWarrior for HC08 V5.0, you might want to add the Build Target combo Box in a project created with V5.0. To do so follow these steps:

- Select “Edit” -> “Commands and key Bindings”, The “Customize IDE Commands” dialog is opened.
- Switch to the “Toolbar Item” Tab
- Scroll down until you see “Current Target” icon.



- Click on the “Current Target” icon, keep the mouse button pressed and drag the icon over to your project window toolbar. You will see a black line indicating where the icon will be placed on the tool bar.
- Release the mouse button when you have reached the desired position.
- The combo box is available now.

