

FAQ CodeWarrior for 8 & 16-bit Tools – Technical Support

Table of Contents

FAQ CodeWarrior for 8 & 16-bit Tools – Technical Support.....	1
Table of Contents.....	1
FAQ Installation	2
FAQ Licensing.....	2
FAQ Batch Mode – Command Line Tools.....	4
FAQ IDE.....	4
FAQ Project Management	7
FAQ Project	8
Project HC05 and ST7	14
Project HC08.....	15
Project HC16.....	15
FAQ Assembler	16
Assembler HC05, ST7	17
Assembler HC12.....	17
FAQ Compiler	18
HC12 Compiler.....	20
HC05 Compiler.....	21
ST7 Compiler.....	21
FAQ Libraries	22
HC12 FAQ Libraries	23
FAQ Linker.....	23
FAQ Debugger.....	24
Mon08	29
SDI.....	29
Simulator HC11	30
Small Monitor (HC11).....	31
ICD HC16	31
Simulator.....	31

FAQ Installation

Q *How can I access the Technical Notes about 8-16 bits tools?*

A When you select the typical installation for 8-16 bits tools, technical notes are not installed automatically. You can install them on your Machine.

?? In the Start Menu select "Setup or uninstall" in the program group the installer has created.

?? Select Modify installation.

?? In the component window check the Technical Notes group and continue the installation.

?? Technical notes are then installed in the installation "docu" or "CodeWarrior Manuals" directory.

Q *How can I retrieve the version number of the CD I have installed?*

A There are several ways to figure out which is the version number from an 8-16 bit CD.

?? In the installation directory you can find a file called "Welcome_*.txt". The CD version number is specified at the very beginning of this text file.

?? On the installation CD, take a look at the file versions.pdf in the directory "HIWARE\Docu". The current CD version number is specified on top of the file.

Q *I am running Windows NT or Windows2000 and I am not able to install CodeWarrior?*

A Make sure to login as an Administrator when you install the tools.

FAQ Licensing

Q *I am using another software, which is licensed using FLEXlm. When I merge the Metrowerks license file with the other vendor license file, all my CodeWarrior tools are working in demo mode.*

A In fact you can define a different service for each vendor using FLEXlm. This way you can work in parallel with products using different version of FLEXlm.

?? On the server define two different directories to store the FLEXlm files, one for the Metrowerks license management and another one for the other vendor license management.

?? Split the license file. Use one for the Metrowerks product and another one for the other vendor products.

?? Make sure to use different port number in the two license files.

?? Install two different FLEXlm services, one for each software you are using.

1. Start the FLEXlm License manager from the Control Panel

2. In the Setup Tab, define a "Service Name" Metrowerks FLEXlm License manager

3. Initialize "lmgrd.exe" with the path and name of the Metrowerks license manager program.

4. Initialize "License File" with the path to the Metrowerks server license.dat file.

5. Check "Use NT Service" or "Start Server at Power Up" depending if you are using an NT or 95 OS on your license server.

6. Go to the Control tab and press "Start" to start the service.

7. In the Setup Tab, define a "Service Name" xxx FLEXlm License manager, where xxx stand for the vendor name.

8. Initialize "lmgrd.exe" with the path and name of the other vendor license manager program.

9. Initialize "License File" with the path to the vendor license.dat file.
10. Check "Use NT Service" or "Start Server at Power Up" depending if you are using an NT or 95 OS on your license server.
11. Go to the Control tab and press "Start" to start the service.
12. Reboot the server.

Q *I got a correct license file, but my 8-16 bit tools keeps telling me I am in demo mode.*

A This may happen when the tools are not looking for the license file in the appropriate directory.

?? In the PC Start Menu select "Run". The Run dialog is opened.

?? In the "Open" edit box type regedit and click "OK". The Registry editor is opened.

?? Zoom in "HKEY_LOCAL_MACHINE".

?? Zoom in "Software"

?? Select "FLEXlm License Manager". On the right of the window you see an entry "HIWARE_LICENSE_FILE".

?? Select the entry "HIWARE_LICENSE_FILE" and select "Edit|Delete" to remove the entry from the registry.

?? Close the Registry editor.

Q *I am already using 3 floating licenses for one CPU (for example HC08) and I intend to order 2 floating licenses for another (for Example HC12). I want to use the same server for both licenses. What should I care about?*

A Following points must be taken into consideration:

?? If the two licensing scheme are using two different daemons, define one FLEXlm service for each license.

?? If the two licensing scheme are using the same daemon, you should have one single FLEXlm service handling both licenses. License information for the two toolsets must be merged in a single license file.

Be careful number of licenses for the tools, which are included in both toolsets, must be equal to the sum of the licenses for the first toolset and the ones for the second one,

If you intend to order 3 HC08 licenses and 2 HC12 licenses, the license file should contain:

?? 3 licenses for HC08 compiler

?? 2 licenses for HC12 compiler

?? 5 licenses for SmartLinker

?? And so on,

Q *If my client dies, does the server free the license?*

A Yes, unless the client's whole system crashes. Assuming communications is TCP, the license is automatically freed immediately. If the whole system crashes, then the license is not freed, and you should use lmremove to free the license.

A If the whole system running the client application crashes or is disconnected before the license is checked in, the license will NOT be freed automatically. (This is most common on a PC.) To free the license, use lmutil lmremove or stop and restart the server.

FAQ Batch Mode – Command Line Tools

Q *I want to use the CodeWarrior build tools in Batch mode (from a batch File or from a make file)*

A Take a look at technical Note [TN 100](#).

Q *I have installed my license file in the root installation directory, but the compiler, linker keeps telling me I am working in demo mode.*

A When you are using the tools in batch mode (from a batch file, a make file or from the command line), you have to install the license file in the PROG installation directory. This is the place where the EXE files are stored. Note: Since 2002 the tools also supports the license file located in the installation directory (that is above the 'prog' directory, the same as the one used by IDE). Have a look into the about box of compiler/linker/ ... to check which license file is used

Q *How can I get the output from the tools redirected to the PC standard Output (STDOUT)*

A Start the tool through the piper.exe tool. Command line for the linker can look as follows:

```
N:\MW\prog\piper.exe N:\MW\prog\linker.exe fibo.prm -Prod=N:\MW\Work
```

FAQ IDE

Q *Starting up the IDE (from the Start menu or directly by clicking on the IDE.exe file), it takes about 5 to 10 seconds before Windows crashes. Running the regservers.bat file before doesn't fix the problem.*

A The problem may be generated when the "My documents" folder is located on a subst drive. Moving the folder "My Documents" to the C drive ("[C:\My Documents](#)") may solve the problem. This problem is mainly reproducible on Host running a non-English OS version.

Q *How can I store project information in an ASCII text File?*

A Use menu entry "File" -> "Export" to store project information in an .XML file. XML file have extended HTML format.

Q *How can I select a rectangular block of text?*

A In order to be able to select a vertical block process as follows:

?? Press the <ALT> key and keep it down

?? Click and drag the cursor to select a rectangular portion of text.

?? Once the vertical block is selected you can release the mouse button and the <ALT> key. The rectangular text selection is highlighted.

Q *How can I create new menu entries in the IDE menu bar?*

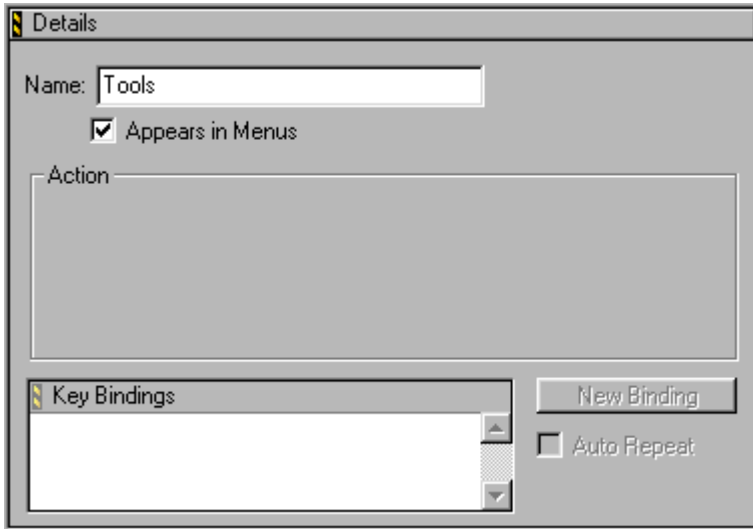
A Use menu entry "File" -> "Commands and Key Bindings" to add new menu or menu entries.

1. Adding new menu:

?? In the "Customize IDE Commands" dialog, click on the "New Group" button.

?? In the Detail dialog on the right hand side, specify the name of the new menu in the "Name" edit box.

?? Click on “Save” to add the new menu in the menu bar.



2. Adding new menu entry

?? In the “Customize IDE Commands” dialog, select the menu where you want to add a new entry.

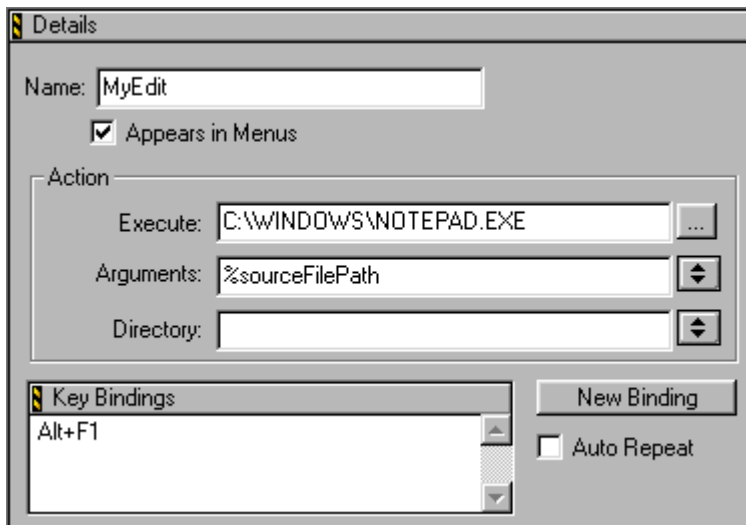
?? Click on the “New Command” button.

?? In the Detail dialog on the right hand side, specify the name of the new menu entry in the “Name” edit box.

?? Browse for the executable file or batch file you want to associate to the menu entry in the “Execute” edit box.

?? Execute the executable or batch file arguments in the “Arguments” edit box.

?? Click on “Save” to add the new menu in the menu bar.



Q *I have set some breakpoints in the Editor window. These breakpoints are ignored when I start the debugger. Why is that happening?*

A Breakpoints set in the editor are not transferred to the debugger. This is due to the fact that we are using an external debugger in CodeWarrior for 8-16 Bit tools.

FAQ Project Management

Q *I want to use CodeWarrior to debug an existing executable file. I do not want to rebuild the application.*

A Take a look at technical note TN 203.

Q *How can I specify the sequence in which the source files are processed while building an application?*

A The “Link Order” Tab in the Project Window allows you to specify in which sequence the files must be processed during the build session.

Q *I want to use a debugger from an Emulator vendor to debug my application.*

A This can be configured in the Build Extra Panel.

?? Open the Project Settings dialog.

?? Select the “Build Extras” panel.

?? Make sure “Use External Debugger” is checked.

?? In the “Application” edit box, browse to the debugger executable file name.

?? In the “Arguments” edit box, specify the options you have to use to start the debugger. Contact your emulator vendor for information about the requested options.

?? In the “Initial Directory” specify the working directory.

?? In order to set up the debugger environment, refer to the debugger user manual.

Q *How can I start another tool from CodeWarrior*

A There are different ways to integrate third part tools in the CodeWarrior IDE:

1. Check with the tool vendor if a plug-ins is available to integrate the tool into CodeWarrior. If a plug-ins is available, install it according to the instruction from the tool vendor.

2. When no plug-ins is available you can start the tools through command line argument, take a look at the technical note TN 202.

Q *How can I use a Version Control Software in CodeWarrior*

A In order to be able to use a version Control Software (VCS), you have to install the corresponding plug-ins.

1. Go to the Metrowerks Web site www.metrowerks.com and check if a plug-ins is available there for the VCS software you intend to use.

2. If a plug-ins is available there, download it and install it according to the installation guidelines.

3. If there is no plug-ins available on the Metrowerks web site, check with your CVS software vendor if he has a plug-ins for CodeWarrior. If a plug-ins is available there, install it according to the installation guidelines.

4. Once the plug-ins has been installed, you have to activate it for the project. In this purpose:

- ?? In the “Edit” menu select “Version Control Settings ...”. The VCS Settings dialog is opened.
- ?? In the CVS Setup Panel, make sure the “User Version Control” check box is checked.
- ?? In the “Method” combo box, select the appropriate VCS software.
- ?? For configuration of the plug-ins, refer to the VCS plug-ins user manual.

FAQ Project

Q *How can I generate a Motorola S file for my application*

A A Motorola S file is automatically generated at the end of the build process. This file is stored in the output directory and has extension .s19. If no Motorola S file is generated in the bin directory, make sure a file with extension .bbl is inserted in the project. The .BBL file contains burner commands used to generate the Motorola S File. For more information about the available commands take a look at the Burner User Manual Chapter “Batch Burner” section “Parameters of the Command File”.

Q *When I try to build my project, I get the message "ERROR C5200: ----- file not found"?*

A This happens when the paths to your include files are not specified in the project Access Paths.

?? In order to add paths where to find the include files, follow these steps:

1. Open the Project setting dialog box.
2. Select the "Access Paths" panel.
3. If your include file are specified in double quotes ("xxx.h"), check the "User Path" option button. If your include file are specified in brackets (<xxx.h>), check the "System Path" option button
4. Click on the "Add" button. The “Please select an Access path” dialog is opened.
5. In the “Path Type” combo box, specify if you want to define the path relatively to your project (“Project Relative”), to your CodeWarrior installation (“Compiler Relative”), to the Windows directory (“System Relative”) or using an absolute path.
6. Browse to the directory you want to include in your access path and click “OK”. Be careful, two folders displayed to the left-hand side of the path name indicate that the directory itself and all its sub-directory are inserted in the Access Paths.

?? If your included files are located in several directories, repeat steps 3 to 6 above for each of them.

Q *My project works without any problem in the debugger and does not work stand-alone?*

A When trying to generate a stand-alone application, check following points.

- ?? The reset vector must be initialized in the ABS file.
- ?? Make sure that the COP (or watchdog) is disabled.
- ?? Make sure that all the registers initialized in the files “startup.cmd” and “reset.cmd” are initialized in the application startup code.

Q *You don't want to use the CodeWarrior IDE; you want to use another editor.*

A To use another IDE, you must integrate the tools available on the CodeWarrior CD in the IDE. To help you to do this, there are several technical notes available on your CD or must be download on our Web site:

- ?? TN17.pdf : Integration into Microsoft Visual Studio (Visual C++ V5.0 or later)

- ?? TN91.pdf : Using CodeWright V5.0 as an IDE
- ?? TN94.pdf : Using UltraEdit V6.1 as an IDE
- ?? TN96.pdf : Using WinEdit98 or latter as an IDE
- ?? TN98.pdf : Using WinIDEA as an IDE
- ?? TN99.pdf : Using DA-C as an IDE.
- ?? TN73.pdf: Using a third party Editor or IDE to start the Metrowerks Tools

Q *I am implementing an application for a MCU, and there is no stationery available for this MCU in my CodeWarrior installation. How can I create my own stationery?*

A It is very easy to create stationery for a MCU.

- ?? Create a new project for an existing MCU, using the same target interface you intend to use for your own project. Name it according to the new MCU
- ?? Open the PRM file(s) inserted in the project and adjust it to your MCU. Mainly modify the memory areas defined in the SECTIONS block to match your hardware.
- ?? Save the file.
- ?? Start the Debugger and adjust the settings until you can successfully connect to your target board. Do not forget to store your settings using “File”-> “Save Configuration” in the debugger.
- ?? Now close CodeWarrior.
- ?? Copy the whole project directory to the Stationery installation directory. Place it in the appropriate CPU and target interface directory.
- ?? Delete the content of the bin subdirectory.
- ?? Remove the directory <project>_data in the stationery.
- ?? Next time you start CodeWarrior, the stationery will be available in the New Project dialog box.

Q *When I try to create a project from a stationery, 10 min. are required until the project window (<project>.mcp) is opened*

A It is possible that the IDE is looking for a license file in a place, which is currently not accessible. It may try to connect to an external floppy drive or to another PC while your station is off line.

- ?? In the PC Start Menu select “Run”. The Run dialog is opened.
- ?? In the “Open” edit box type regedit and click “OK”. The Registry editor is opened.
- ?? Zoom in “HKEY_LOCAL_MACHINE”.
- ?? Zoom in “Software”
- ?? Select “FLEXIm License Manager”. On the right of the window you see an entry “HIWARE_LICENSE_FILE”.
- ?? Select the entry “HIWARE_LICENSE_FILE” and select “Edit|Delete” to remove the entry from the registry.
- ?? Close the Registry editor.
- ?? Now creating a new project from stationery should go faster.

- Q** *While linking my application, I get a message “L1818: Symbol 16 - _Startup duplicated in starts.o and start.c.o”*
- A** The symbol _Startup is detected twice in the application. This mainly happens when the startup file is inserted in the project and also specified in the NAMES block of the PRM file. In order to solve this you can:
- ?? Modify the PRM file and remove the startup file from the NAMES block or.
 - ?? Remove the startup file from the application.
- Q** *While linking I get a message “L1822: Symbol <SymboleName> in file <FileName> is undefined”*
- A** The specified function is invoked in the specified module, but is not implemented anywhere in the application. In order to solve this problem:
- ?? If the function is an ANSI library function or a run time function, add the ANSI library file to your application. Usually runtime functions names start with an underscore (‘_’).
 - ?? If the function is a user-defined function, add the source file where the function is implemented to the project.
- Q** *I have a very short application. I want the assembler to generate directly an executable file. I do not want to use the Linker*
- A** Take a look at technical note TN 204.
- Q** *How can I link data to an application*
- A** Take a look at the technical note TN59.
- Q** *What to do if an assertion happens (internal error)?*
- A** Extract source, where the assertion appears, and sends it as zipped file with all headers, used options and versions of all used tools.
- Q** *How can I retrieve the version number from a Compiler, Assembler or Linker in my CodeWarrior environment*
- A** This can be retrieved from the Project setting dialog.
- ?? Open the Project setting dialog
 - ?? Select the appropriate tool panel (For example the Compiler Panel)
 - ?? Press the “About” button. A dialog opens containing all version information about the tool. Content of this dialog can be copied to the clipboard and sent over to support.
- Q** *When building the application, the IDE is trying to access the drive A:. All my source files are located on my disk drive.*
- A** It is possible that the IDE is looking for a license file on drive A:
- ?? In the PC Start Menu select “Run”. The Run dialog is opened.
 - ?? In the “Open” edit box type regedit and click “OK”. The Registry editor is opened.
 - ?? Zoom in “HKEY_LOCAL_MACHINE”.
 - ?? Zoom in “Software”
 - ?? Select “FLEXIm License Manager”. On the right of the window you see an entry “HIWARE_LICENSE_FILE”.

- ?? Select the entry “HIWARE_LICENSE_FILE” and select “Edit|Delete” to remove the entry from the registry.
- ?? Close the Registry editor.
- ?? Now the IDE should not make any attempt to access the drive A:.

Q *How can I tell the IDE to ignore specific directory when building an application?*

A Directories, which should not be touched while building an application, must be specified in the “Shielded Folders” panel. This may be useful to skip directories created by VCS software inside of the project directory during building of an application.

- ?? In the “Edit” menu select “Preferences”.
- ?? Select the “Shielded Folders” panel on the left hand side of the “IDE Preferences” dialog.
- ?? In the “Regular Expression” edit box specify the directory (or directories) you wish to ignore when building an application. Take a look at the IDE User Manual chapter “Finding and Replacing Text”, section “Regular-Expression Find” for information about valid notation for regular expressions.
- ?? Make sure the “Project operations” checkbox is checked.
- ?? Click on the “Add” button to add the specified directory to the list of shielded folder.

Q *I am using library or an SDK from a third party company inside of my project. How can I make sure my project is easily portable when referring to the third party SDK or library*

A This can be performed using Source trees. Source Trees may be defined globally (for all projects on the PC) or for a specific project only.

- ?? To define a source tree for all projects on the PC, in the “Edit” menu select “Preferences”. Select the “Source Trees” panel on the left hand side of the “IDE Preferences” dialog.
- ?? To define a source tree for a specific project, open the project settings dialog. Select the “Source Trees” panel on the left hand side of the project settings dialog.
- ?? Enter an identifier you will use to refer to the third party libraries or SDK.
- ?? In the type combo box, you can specify if you want to use an Absolute Path, an Environment Variable or a Registry Key to define the source tree.
- ?? In the edit box, specify the Absolute Path, Environment Variable or a Registry Key name.
- ?? Click on the “Add” button to add the specified source tree to the list of available source tree. From now on you can use the source tree name to refer to the library or SDK installation directory. Suppose you have defined a Source tree called “TPLib”, you can use {TPLib} to refer to the source tree.
- ?? This makes your project more portable when transferring the project to another PC. Just define the Source tree accordingly on the new PC.

Q *How can I check the assembly code generated by an ANSI C source file?*

A In order to get the disassembly listing for a C source file:

- ?? Click with the right mouse button on the source file name in the project window
- ?? Select “Disassemble”.
- ?? The disassembled code is displayed in an edit window. Use the standard Save or Save as menu entry in the IDE to save the files on the disk.

Q *How can I see the number of execution cycles of the generated code?*

A This information can be retrieved for each source file in the application. You have to use the decoder and to activate the option -T. In order to achieve that:

?? Open your project setting dialog.

?? Click on "Importer to <ChipName>".

?? In the "Command Line Arguments" edit box enter -T.

?? Click on "Compiler for <ChipName>".

?? Make sure the "Use Decoder to generate Disassembly Listing" box is checked.

?? Make sure the option -iasm is not specified in the "Command Line Arguments" dialog box.

?? Now in order to get the disassembly listing for a C source file, click with the right mouse button on the source file name in the project window and select "Disassemble".

?? The disassembled code (including cycle count) is display in a window. Use the standard Save or Save as menu entry in the IDE to save the files on the disk.

Q *When Attempting to link my application I get the following message: " Link Error: L2075: 8 bit index out of range (561 for _<SymbolName>)". How can I avoid this message?*

A Message L2075 is usually generated when a symbol (variable) is defined in a SHORT section but allocated outside of the direct page (0x00 .. 0xFF) by the linker. Check following point:

?? The name of the symbols generating the problem is specified in the error message. Check this variable is definition and declaration. Make sure the variable is defined and declared in the same section, using the same section attribute.

Q *I want to insert a copyright string in my binary files. How can I achieve that?*

A Copyright string can be defined through the environment variable COPYRIGHT. In order to define a copyright string, specify following option in the Compiler, Assembler or Linker command line option:

`-Env"COPYRIGHT=This is my Copyright".`

A Copyright string is then stored in the binary file.

?? When you are working in ELF DWARF format, the COPYRIGHT string is stored in the .note section. Decode the abs file with option -e -d.

?? When you are working in HIWARE file format, the COPYRIGHT string is displayed in the "Header Information". Decode the abs file with option -a.

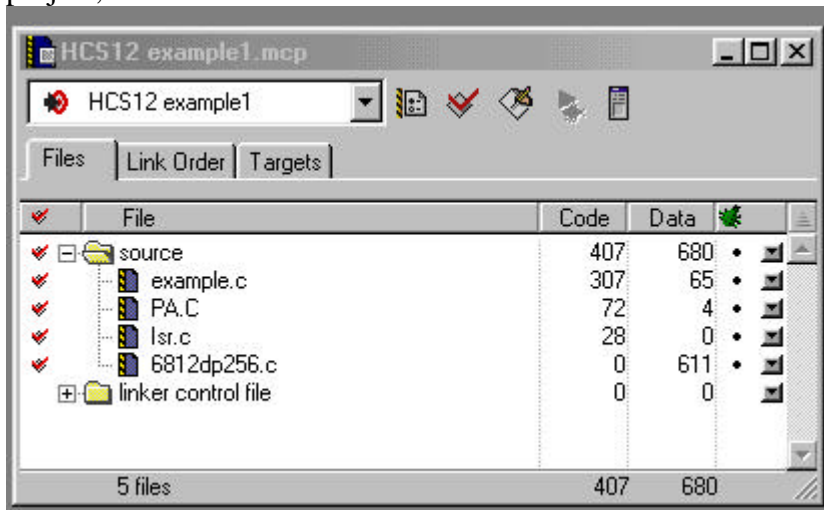
Q *My application works while debugging but not in stand-alone.*

A In most of the hardware the watchdog (or COP) is disable by default when you enter the debug mode (mon08, debug via bdm port...). When you are starting your application stand-alone, you have to disable it explicitly. Make sure that your application manage, refresh or disable the Watchdog (or COP).

Q *My executable file does not contain any debug information. How can I make sure debug info are inserted in the abs file?*

A This is due to an incorrect setting in your project. This happens when you create the project from scratch (from an empty project). We do not recommend this approach, because the project may not be set up appropriately. We hardly recommend you to create your project from Stationery. This will save

you a lot of time and avoid a lot of problem. In order to have debug information generated for your project, make sure there is a dot after each file name in the column with a green bug on it.



In order to get the dot there, click with the mouse in the green bug column in the line where the Sources group appear.

Q *How can I initialize Bytes of Flash or EPROM, which are not used by the application, with a fill pattern?*

A You can use the modifier FILL in your PRM file SECTIONS definition in this purpose.

Example:

```
SECTIONS
    Z_RAM = READ_WRITE 0x0080 TO 0x00FF;
    MY_RAM = READ_WRITE 0x0100 TO 0x01FF;
    MY_ROM = READ_ONLY 0xF000 TO 0xFEFF FILL 0xFF;
```

Q *I want to compile some source code inside of my application with different options. How can I achieve that?*

A Take a look at technical note TN206.

Q *How can I disassemble a binary executable file?*

A After you have built your project, a binary file is created. This file is an ELF DWARF executable file and can be loaded in a debugger. Usually, this file has the extension .abs. But the programmer can decide to use another extension. In order to disassemble this binary file it should be added to the project. In this purpose:

?? If the extension from this binary file is not .abs, first add this extension in the file mapping.

Take a look at Technical Note TN201 for information on how to add file extension to the File Mapping Panel.

?? When adding extension for an executable file, make sure that the field are initialized as follows:

?? Set "Compiler" to "??? Importer", where ??? stands for the CPU you are targeting..

?? Set "Edit language" to "None".

?? Now, to disassemble the executable file,

- ?? Add it in your project (Project | Add).
- ?? Click with the right mouse button
- ?? Select Disassemble in the pop up menu.

Q *I want to debug the abs file I have generated previously. I do not want to rebuild the ABS file prior to starting the debugger. How can I achieve that?*

A If you want to load your application in the debugger without rebuilding it, follows these steps:

- ?? Start CodeWarrior IDE
- ?? Select "Edit" -> "Preferences". The IDE Preferences dialog is opened.
- ?? Activate the "Build Settings" panel.
- ?? Set "Build Before running" to "Never"
- ?? Click OK to close the IDE Preferences dialog.

Be careful this setting applies to the whole Host PC. This means it will apply to all projects developed on the PC.

Project HC05 and ST7

Q *How can I configure the location of the vector table for the HC05 or ST7 MCU*

A This can be defined through the environment variable RESETVECTOR.

- ?? Depending on the CPU derivatives, the reset vector is not always located at the same address. The environment variable RESETVECTOR defined in the Debugger project file specifies the location of the reset vector. The other vectors are located above this address.
- ?? Edit the Debugger Project File.
- ?? In the section “[Environment Variables]”, add “RESETVECTOR=0x3FFE” to indicate that the reset vector is located at address 0x3FFE.
- ?? The RESETVECTOR environment variable is then used by
 - ?? The simulator to retrieve the PC after Reset,
 - ?? The Compiler to initialize the vector table when an interrupt is defined as “interrupt 3 void myInt(void)”.
 - ?? The Linker, to initialize the vector table, when following command is used: “VECTOR 3 myInt”.

Q *How can I configure the location of the stack for the HC05 or ST7 MCU*

A This can be defined through the environment variable RESETSTACK.

- ?? Depending on the CPU derivatives, the stack pointer after reset does not always point to the same address. The environment variable RESETSTACK defined in the Debugger project file specifies the location of the reset stack.
- ?? Edit the Debugger Project File.
- ?? In the section “[Environment Variables]”, add “RESETSTACK=0x3FF” to indicate that the reset vector is located at address 0x3FF.
- ?? The RESETSTACK environment variable is then used by
 - ?? The simulator to initialize the register SP after a reset.

Q *When Attempting to link my application I get the following message:” Link Error: L2075: 8 bit index out of range (561 for _<SymbolName>)”.* **How can I avoid this message?**

A Message L2075 is usually generated when a symbol (variable) is defined in a SHORT section but allocated outside of the direct page (0x00 .. 0xFF) by the linker. Check following points:

- ?? If you are working in small memory model (-Ms), check in your PRM file if all the data sections and the section `_OVERLAP` are placed on the direct page (between 0x00..0xFF).
- ?? If you are working in large memory model (-Ml), check in your PRM file if the sections `_OVERLAP` and `_ZEROPAGE` are placed on the direct page (between 0x00..0xFF).
- ?? If you are working in extended memory model (-Mx), check in your PRM file if the section `_ZEROPAGE` is placed on the direct page (between 0x00..0xFF).
- ?? The name of the symbols generating the problem is specified in the error message. Check this variable is definition and declaration. Make sure the variable is defined and declared in the same section, using the same section attribute.

Project HC08

Q *How to minimize code size for a HC08 application?*

A Check following points:

- ?? Check if you need the full startup code: e.g. if you do not have initialized data, you can ignore/remove the copy-down, if you do not need to have the RAM memory initialized, you can remove the zero-out. And if you do not need both, you can remove the complete startup code, and set up your stack in your main routine directly. Set with INIT main in the prm file your main routine as application startup/entry point.
- ?? Check the compiler options: You may use e.g. the `-OdocF` function. This increases compilation time, but decreases code size. E.g. you can try `-OdocF="-or"`. Using the option `-Ll` to write a log file tells you the effect for each single function.
- ?? Check if you can use IEEE32 for both float and double.
- ?? Have a look into the map file: check runtime routines out of `rtshc08`. Check for 32bit integral routines (e.g. `_LADD`, typically they have a `_L` prefix for LONG. check if you really need long arithmetic.
- ?? Enumerations: if you are using enums, by default they are encoded on 16bit. They can be set to unsigned 8bit (see option `-T`, or use `-TE1uE`)
- ?? Check if you are using switch tables. There are options to configure this (see `-CswMinSLB` and others)
- ?? Finally the linker has an option to overlap ROM areas (see `-COCC` option in the linker)

Project HC12

Q *While linking an application, I get the message “L1907 Fixup overflow in <Object>, type <objType> at offset <Address>” on a function. How can I fix this?*

A This may happen when a code section is allocated over two non contiguous memory area as in the example below:

SECTIONS

```
RAM = READ_WRITE    0x0200 TO 0x07FF;
FEE20 = READ_ONLY   0x1000 TO 0x3FFF;
FEE28 = READ_ONLY   0x5000 TO 0xFEFF;
EEPROM = READ_WRITE 0x0C00 TO 0x0FFF;

END
```

PLACEMENT

```
DEFAULT_ROM, ROM_VAR, STRINGS      INTO  FEE20, FEE28;
DEFAULT_RAM                        INTO  RAM;
```

END

Per default the compiler tries to optimize JSR into BSR whenever possible to save some memory. As soon as a function invoked is implemented in the same module and in the same code section as the invoking function and the offset to the called function is between -128, 127, the compiler will issue a BSR, because he assume that the functions will be allocated in the sequence they are implemented. When you define two block of memory for the code sections, the caller and the callee may be allocated in two different blocks and this generates the error.

Usually if you define several memory areas (SECTIONS) in the PRM file and when you are using something like

```
DEFAULT_ROM, ROM_VAR, STRINGS      INTO  FEE20, FEE28;
```

for a code section we hardly recommend to switch the BSR optimization OFF for the application.

You can achieve that activating the option -Onf in the project. Note that in this case you also have to regenerate the ANSI libraries with option -Onf to make sure this does not happen in the libraries.

Take a look at TN105 for information on how to rebuild ANSI library.

Project HC16

- Q** *The HC16 core supports a memory model, where data and program spaces can be separated into two different 1 Meg spaces. Do the compiler and linker support this?*
- A** On the HC16 you should never distinguish between code and data space! If you do that, constants in ROM cannot be accessed anymore!! => So there will only 1 Mbytes to be addressed in total.

FAQ Assembler

- Q** *Why can't I use DEFAULT_ROM and DEFAULT_RAM as section names in assembly language?*
- A** Those names are names of predefined sections for the linker. We do not recommend usage of predefined sections in assembly or C source files. Predefined section names are: DEFAULT_RAM, DEFAULT_ROM, FUNCS, ROM_VAR, STRINGS, COPY, and STARTUP.
- Q** *How can I tell the Assembler that a variable must be accessed using the direct addressing mode?*
- A** Take a look at technical note TN 86.
- Q** *How can I define I/O ports in Assembler?*
- A** Take a look at technical note TN 112.
- Q** *How can I mix ANSI C and Assembly source files in an Application?*
- A** Take a look at technical note TN15 or at the chapter "Mix C and Assembler Application" in the assembler user manual.

Q *I want to defined labels I intend to use as index to access data stored on the stack. How can I do this?*

A Using the Metrowerks assembler, we recommend to use the OFFSET directive to achieve that.

Q *How can I pass an argument containing a coma to a macro?*

A Usually, according to the Assembly Language Input Syntax, the coma is used to separate arguments in assembly instructions or macro invocation. In order to pass an argument including comma to a macro, just enclose the argument in '<' and '>'. Example:

```
LoadD:      MACRO
             LDD    \1
             ENDM

Entry:

             LoadD <0,X>
```

Is expanded as

```
+          LDD    0,X
```

Assembler HC05, ST7

Q *I have an application, where I have mixed Assembly and C source file. How can I define function parameters in the OVERLAP section in assembler?*

A Take a look at TN108.

Q *In an assembly source file I want to simulate function local variables. How can I achieve that?*

A Note that usage of local variables is not very common in assembly language. When you want to allocate a variable in the OVERLAP section, you have to specify in which assembly code section the variable is used. Basically local variables will have section scope, so in order to optimize allocation in the OVERLAP section, you have to make sure each "function" is stored in a separate section. In ELF DWARF file format, the naming convention for local variables is:

```
__OVL_<sectionName>_<localVariablename>
```

For example, suppose you want to define local1 and local2, which are used in code section func1.

Variable needs to be defined in the following way:

```
_OVERLAP SECTION

__OVL_func1_local1: DS.W 1 ;
__OVL_func1_local2: DS.W 1 ;
```

Note that local variables must not be exported using XDEF directive. Additionally, the assembler will not generate any message when __OVL_func1_local2 is used outside of "func1". You have to make sure this is not happening.

Assembler HC12

Q *Assembly generates some warnings (A12003 at all and only issue A1415). How can I avoid this?*

A For the source code below

```
XDEF Entry
```

```
DataSection:    SECTION SHORT
toto    equ    $80
```

```
CodeSection:    SECTION
Entry:
```

```
    ANDB #~toto
    BRA    Entry
```

The assembly generates 2 warnings

```
Warning : A12003: Value is truncated to one byte
```

```
Warning : A1415: Cutting fixup overflow
```

- A The message A1415 does also work/considers forward references, it detects that the argument of ANDB # is unsigned. As operator ~ is applied to the operand, the assembly evaluates the result on 32-bits. However B register is only 8 bits wide. So the assembler generates the messages.

As workaround, use:

```
    ANDB #LOW(~toto)
```

OR

```
    ANDB #(~toto) & $FF
```

FAQ Compiler

Q *How can I tell the compiler to use direct addressing mode to access a variable?*

- A Just define the variable in a SHORT segment in the following way:

```
#pragma DATA_SEG SHORT myShortData
int myVar;
#pragma DATA_SEG DEFAULT
```

Q *I have defined a variable in a short segment, but the compiler does not use direct addressing mode to access it.*

- A Make sure that the variable is also placed in a short segment in its external declaration.

Q *I have an I/O port, which requires to be initialized High Byte first. How can I ensure that the port is initialized in the proper sequence?*

- A Take a look at the Technical Note TN107.

Q *How can I access single bits in an I/O register symbolically in my ANSI C application?*

- A Take a look at the Technical Note TN106.

Q *How can I define my I/O registers in a C module?*

- A Take a look at the Technical Note TN95.

Q *Sometimes functions are placed at a specific address, but you don't have the sources or information about it. You just know e.g. that the function starts at address 0x1234 and you want to call it.*

- A Without having the definition of the function, you will run into a linker error, because you don't have the target function code.

What to do? The solution is to use a constant function pointer to be used:

```
void (*const fktPtr)(void) = (void(*) (void))0x1234;
```

```
void main(void) {
```

```
    fktPtr();  
}
```

With this you get very efficient code and no linker error at all. But you have to guarantee that the function at 0x1234 really exists

Q *I have defined some constants in a C++ source file and I have declared them in another source file. After link the constants are allocated in RAM and the linker generate the message: "Warning L1823: External object xxx in yyy.obj created by default".*

A In C++, constants, which are defined in a module, but are not declared previously, are supposed to be static. An external declaration must precede the constant definition.

Q *How can constant objects be allocated in ROM?*

A Use the modifier const to define the object and make sure to compile your source files with option -Cc. Example:

```
const int myConst = 0x56;
```

Q *How can constant pointer objects be allocated in ROM?*

A Use the modifier const to define the object and make sure to compile your source files with option -Cc. Example:

```
int * const myConst = 0x56;
```

Q *Why can no initialized data be written into the ROM area?*

A The qualifier const must be used and the source must be compiled with option -Cc

Q *How can I mix ANSI C and Assembly source files in an Application?*

A Take a look at technical note TN15 or at the chapter "Mix C and Assembler Application" in the assembler user manual.

Q *How can I initialize an I/O port, which is located in ROM or Flash in my application?*

A This can be done using the @operator. The following statement allocate the variable MOR at address 0xFF00 and initialize it with the value 0x10:

```
const MOR @0xFF00 = 0x10;
```

Q *When I build my application, the compiler is not able to find header file included as #include <header.h>. How can I fix this?*

A According to the ANSI C standard, <> is used to include system header files, whereas user header files are specified in "". Project paths can be specified in the Access Path Panel from project settings. In order to add a new path:

?? Open the project.

?? Open the project setting dialog. This can be done hitting ALT+F7 on the keyboard or clicking on the project setting icon in the mcp window.

?? Go to the Access Path panel.

?? Select the "System Paths" radio button.

?? Now using the Add button you can browse for the additional system path directory (where the file header.h is located).

HC12 Compiler

Q *The compiler generates superfluous code in the interrupt function entry and exit code. In the entry code, the content of address 0x35 (=53) is pushed on the stack. In the exit code a byte is popped from the stack and written to address 0x35.*

- A** You are currently working in banked memory model and the compiler must save the value of the PPAGE register in the interrupt entry code and restore the initial value at the end of the interrupt.
- ?? Per default the compiler supposes the PPAGE register is located at address 0x35 (PPAGE location of HC12A4). If the PPAGE register is located somewhere else, you should inform the compiler using the option -CpPPAGE.
 - ?? To tell the compiler the PPAGE register is located at address 0xFF, add -CpPPAGE=0xFF to your command line option.
 - ?? If you want to set the option using the GUI, check the box "PPAGE register is Used for paging" in the "Code Generation" Tab and specify the address of the PPAGE register in the edit box.

Q *I build my application in banked memory model and I am using banked data. How can I make sure data are accessed using run-time functions and that the content of the PPAGE register is saved in interrupt functions entry code?*

- A** Use option -CpPPAGE=<PPAGE ADDR> to specify the location of the PPAGE register and use option -CpPPAGE=RUNTIME to tell the compiler to use Run Time functions to access banked data. Both options must be specified on the command line and -CpPPAGE=RUNTIME must be specified the last.

Example for HCS12DP256

```
-CpPPAGE=0xFF -CpPPAGE=RUNTIME
```

Q *I want to allocate String Constants into Banked Memory. What should I care about when implementing my Code?*

- A** If you intend to allocate your string constants in banked memory:
- ?? Define your string constants in a user defined string segment with attribute PPAGE. As you want to allocate string constants in ROM, you have to use PPAGE to access them.
 - ?? Make sure to use far pointers rather than pointer to define any variable pointing to such a string.

Example:

```
#pragma STRING_SEG PPAGE myStrConst
char *far myStr = "This is a constant String";
#pragma STRING_SEG DEFAULT
```

- ?? Place your segment in banked memory in your PRM file.
- ?? Use option -CpPPAGE=RUNTIME, to tell the compiler you want to access the string using a runtime function. You cannot directly modify the PPAGE register, because the code accessing the string may be located in another banks.
- ?? Add the file datapage.c (located in the LIB\HC12C\SRC installation directory) to your project. Make sure it comes before ansib.lib in the "Link Order" View.

- ?? Make sure the code in the runtime function is correct. Depending on the MCU you are using, you may have to use a compiler option to activate bank switching for HC12DG128 or HCS12DP256 to select the appropriate page register.
- ?? To activate page switching for HC12DG128, compile with option -DDG128.
- ?? To activate page switching for HCS12DP256, compile with option -DDP256
- ?? If you intend to use some library functions to manipulate the strings you have to regenerate the library for far string pointer. In this purpose:
 - ?? Edit the file libdefs.h located in the installation directory "LIB\HC12C\include". In this file modify lines 74 and 78 as follows:


```
#define LIBDEF_FAR_CONST_STRINGS 1
#define LIBDEF_FAR_STRINGS 1
```
 - ?? Generate the library again. This can be done using the project hc12 lib.mcp located in the installation LIB\HC12C directory. Technical note TN105 provide you information how to generate a new library.

HC05 Compiler

Q *How to avoid reentrancy problems when a function is invoked from the main function and from an interrupt.*

A In order to avoid this I see three solutions:

?? Enclose the invocation of the function between DisableInterrupt .. EnableInterrupt.

```
DisableInterrupts;
function(p0);
EnableInterrupts;
```

?? The point here is it acceptable to disable interrupts during execution of these functions.

?? Inside of the interrupt function, push value from the address where the function local variables and parameters are stored on the stack prior to invoking the function and pop them after the function has been executed.

Suppose the function has a single parameter, which is stored at address 0x61.

```
asm LD A, 0x61
asm PUSH A
function(p0);
asm POP A
asm LD 0x61, A
```

?? Duplicate the function. The main and interrupt functions will call a separate function.

ST7 Compiler

Q *How to avoid reentrancy problems when a function is invoked from the main function and from an interrupt.*

A In order to avoid this I see three solutions:

?? Enclose the invocation of the function between DisableInterrupt .. EnableInterrupt.

```
DisableInterrupts;
function(p0);
EnableInterrupts;
```

- ?? The point here is it acceptable to disable interrupts during execution of these functions.
- ?? Inside of the interrupt function, push value from the address where the function local variables and parameters are stored on the stack prior to invoking the function and pop them after the function has been executed.

Suppose the function has a single parameter, which is stored at address 0x61.

```
asm LD A, 0x61
asm PUSH A
function(p0);
asm POP A
asm LD 0x61, A
```

- ?? Duplicate the function. The main and interrupt functions will call a separate function.

FAQ Libraries

Q *Footprint from function "printf" is too big. How can I reduce the code size for this function?*

A In fact according to the ANSI C standard, the function printf must support a lot of different formats. This is the main reason why this function takes so much memory.

As we know that the amount of memory is limited on an embedded processor, there is a way to minimize the footprint for the function printf.

- ?? Add the file printf.c located in your LIB\HC08c\src installation directory to your project.
- ?? In the "Link Order" view make sure the file printf.c comes before the file ansi.lib. Use drag and drop to bring the file ansi.lib at the end of the list for example.
- ?? Edit the file "libdefs.h" located in your LIB\HC08c\include installation directory.
- ?? In this file, there is a couple of macros, which can be modified to minimize the amount of code in the function vprintf. The macros are defined around line 180.
- For instance setting LIBDEF_PRINTF_FLOATING to 0 will reduce the code drastically. Be careful, with this modification the customer will not be able to format floating point values (float, double) any more.
- Usage of each macro is explained in the header file. So you should be able to adjust the function vprintf to your own needs.
- ?? Now build the application.

Q *I want to redirect STDOUT (printf) from my application to a specific out put channel (SCI, LCD, ...). How can I do this?*

A The low level input/output functions are implemented in the file termio.c located in your installation "LIB/<CPU>C/Src" (Where <CPU> stands for the name of the CPU).

- ?? "TERMIO_GetChar": is the function used to receive character from the terminal channel.
- ?? "TERMIO_PutChar": is the functions used to send a character to the terminal channel.
- ?? "TERMIO_Init": is the functions used to initializes the communication channel.

In order to redirect STDOUT to another output channel:

- ?? First, make a backup of the file termio.c
- ?? Then, add the file termio.c in your project

- ?? Adapt the functions “TERMIO_GetChar”, “TermIO_PutChar” and “TermIO_Init” to the output channel you wish to use.
- ?? In the mcp window of your application, in the 'link order' section, the termio.c must appear before the ANSI libraries (this will make sure the application uses the modified termio.c rather than the one from the library.).

HC12 FAQ Libraries

Q How can I make sure that the ANSI library is able to handle the banked strings constants and banked constants correctly?

- A Rebuild the libraries specifying that all string constants are far.
- ?? Open the project “hc12 lib.mcp”.
 - ?? Edit the file "libdefs.h" located in your LIB\HC12c\include installation directory.
 - ?? Set LIBDEF_FAR_CONST_STRINGS to 1
 - ?? Set LIBDEF_FAR_STRINGS to 1
 - ?? Set LIBDEF_FAR_CONSTANTS to 1
 - ?? Now rebuild the libraries. The new generated libraries are no able to handle far string constants and far constants.

FAQ Linker

Q How can an own startup be called after reset?

- A In the prm file use following syntax to tell the linker that the application entry point is called “myStartup”:
- ```
INIT myStartup
```

***Q How can the reset vector be set to the beginning of the startup code?***

- A In the prm file initialize the reset vector as follows:
- ```
VECTOR 0 _Startup /* set reset vector on _Startup */
```

Q How can I switch OFF smart Linking (dead stripping) in my application

- A Take a look at the manual chapter “Linking Issues”; section “Smart Linking (ELF)” if you are working in ELF DWARF file format.
Take a look at the manual chapter “Linking Issues”, section “Smart Linking (HIWARE+ELF)” if you are working in HIWARE file format.

Q I want to link my application containing absolute assembly sections and the linker keeps reporting overlaps between sections.

- A In fact when you are linking an application containing absolute assembly sections, you have to make sure there is no overlap between the absolute section defined in the assembly source file and the memory area defined in the PRM file.

Q Error “L1907: Fixup overflow in <Object>, type <objType> at offset <Address>” is generated when I try to link my application. How can I avoid it?

A This may happen when “Branch JSR to BSR optimization” is activated and a code section is placed in two non-contiguous memory area in the PRM file as.

```
SECTIONS
    RAM = READ_WRITE      0x0200 TO 0x07FF;
    FEE28 = READ_ONLY     0x1000 TO 0xFCF0;
    FEE32 = READ_ONLY     0xFD00 TO 0xFEFF;
    EEPROM = READ_WRITE  0x0C00 TO 0xFFFF;

END

PLACEMENT
    DEFAULT_ROM, ROM_VAR, STRINGS      INTO  FEE28, FEE32;
    DEFAULT_RAM                        INTO  RAM;

END
```

Per default the compiler tries to replace JSR by BSR whenever possible to save memory. As soon as a function invoked is implemented in the same module and in the same code section as the invoking function and the offset to the called function is between -128, 127, the compiler will issue a BSR, because he assume that the functions will be allocated in the sequence they are implemented. If during linking the calling and called function are allocated in two different memory area, the offset is then bigger than -128 and the linker complains.

In order to avoid this message there are two solutions:

- ?? Allocate the code section in one single memory block.
- ?? Compile the application with option `-Onb=b`, which switched “Branch JSR to BSR optimization” OFF. Note that in this case you also have to regenerate the ANSI libraries with option `-Onb=b` to make sure this does not happen in the libraries. Take a look at technical note TN205 for information on how to generate new libraries.

FAQ Debugger

Q How can I display the communication protocol between the target and the host?

A This can be done checking the “Show Protocol” check box in the Communication dialog.

- ?? Start the Debugger
- ?? Select “Component|Open”. The Open Window Component is opened.
- ?? Select “Command” and click “OK”. A Command component is opened. You can resize and move the window to the appropriate position.
- ?? In the target menu select the entry ”Setup” or “Communication”. The Setup Communication dialog is opened.
- ?? Check the “Show Protocol” check box and click “OK” to close the dialog.
- ?? Select “Project|Save” to save the new project settings.
- ?? Close the Debugger and restart it. The communication protocol is written to the Command Component.

Q *During loading of an application, the message "Error while downloading at address <Address>" is issued.*

A This happens mainly when the debugger does not have any write access to the specified address.

?? Check your hardware board Memory Map and make sure <Address> is in the valid memory range.

?? If the address resides in the target board FLASH EPROM area

?? Check if flash programming is supported by the debugger on your target board or if your target interface supports FLASH Programming.

?? Make sure flash programming has been enabled before you have loaded your application.

?? If the address resides in the EEPROM area for a M68HC12 processor

?? Check if your target interface supports EEPROM Programming

?? Make sure that the environment variables EEPROM_START and EEPROM_END are initialized with the EEPROM start and end address. These variables are defined in the file Debugger Preference dialog. In the debugger select "File|Configuration" and then select "Various Environment Variables".

?? Otherwise

?? Start the Debugger.

?? Open a Memory Component and select 'Address ...' from the Memory Component Pop up Menu

?? In the "Display Address" dialog enter the address in the 'Address' edit box and click 'OK'.

?? Double click on the specified memory address and try to enter some value there. If the memory remains unchanged, you do not have any write access there. The debugger is not able to load an application over there.

?? Check your configuration (Chip Selects, Config Registers, ...)

?? Make sure the address is located in the target board RAM area. The debugger is not able to load an application in EPROM.

Q *I try to set a breakpoint in my source code, but the application does not stop on them. My code is loaded in the CPU Flash memory.*

A This is related to problem setting hardware breakpoints. Hardware breakpoints are not available on all CPU. Please check your target interface manual to detect if they are supported.

?? Make sure Hardware Breakpoints are supported by the target interface you are using.

?? In the target menu select the entry "Set hardware BP". The 'Hardware Breakpoint Configuration' dialog pops up:

?? If you remapped the I/O registers, the hardware breakpoint base address must be set to the address of the Hardware Breakpoint Module after re-mapping. In this case, breakpoints can only be set after the I/Os have been remapped. Do not set any breakpoints in flash after loading the application. Execute application up to the re-mapping of the I/Os before setting any breakpoints.

?? The number of hardware breakpoints available is limited by the hardware. Make sure some hardware breakpoints are free.

Q *How can a loaded program in the debugger be started automatically and stop at a specified breakpoint?*

A In the file postload.cmd add following command to have your application stopping execution in the function main:

```
bs &main t  
g
```

Q *While I am debugging my application, the application source files are not displayed in the Source component.*

A Make sure the paths to all the directories containing your source code are configured in the debugger.

?? Start the Debugger

?? In the File menu select "Configuration ...". The "Preferences" dialog is opened.

?? Add the paths to all the source files inside of your application to the "General Path".

Q *I am debugging an OSEK application and I do not see the OSEK elements in the Inspect component.*

A Check the following points:

?? Make sure that you have a valid license for OSEK awareness in the debugger.

?? Make sure the ORTI file (.ort) is located in the same directory as the absolute file and has the same base file name.

?? After you have loaded the OSEK application, open a command line component prior to opening the Inspect component. Parsing errors in the ORTI file will be written to the command component.

Q *How can I watch only selected symbols in the application?*

A In order to be able to select the symbols or variable you want to watch use the Visualization tool component.

?? Open the Visualization tool component (Component -> Open)

?? Resize the component and place it wherever you want.

?? Add the symbols you want to track in the visualization component

?? For each component repeat these steps:

?? Click right in the Visualization Tool component and select "Add New Instrument"

?? Select "Value as Text". A rectangle will be displayed in the component following the mouse pointer.

?? Click in the Visualization tool component wherever you want to place the instrument. A rectangle displaying "Value: ??????" is displayed.

?? Point to the symbol you wish to add in the Data:1 component drag it to the visualization tool component and drop it in the instrument you have created (rectangle).

?? Click right in the instrument and select "Properties".

?? Change the field "Field Description" and write the name of the variable you have selected there. You can also change other display parameter if you wish (font, color,).

?? Once you have defined all the variables, click right inside of the Visualization tool component (not on any instrument you have defined) and select "Properties".

?? Set "Editmode" to "Off".

?? Select "File" -> "Save Project" in the HI-WAVE menu bar.

This layout will then be reloaded next time you start the debugger. You just have to make it once

Q *How can I quickly find a function used in my application (to set a breakpoint for example)?*

A This can be done quickly using the menu entry Find Procedure or using the Inspector component.

1. Using the "Find Procedure" menu entry.

?? Select the source window,

?? Press right mouse button for the context menu

?? Select 'Find Procedure' or press CTRL-I.

?? This jumps you directly to the function you want, and remembers the function names you type in. You just have to remember the name (and spelling) of the function. This feature is case sensitive.

2. Using the Inspector component.

?? In the menu "Component" select Open,

?? Then select the inspector Component. There is a nice tree control with the full debug information (so contains all the symbolic).

?? Open/unfold it module wise

?? Drag&drop the function names into the source window: it directly jumps you to the source for it!

Q *My application is already loaded into Flash or EPROM. I want to start the debugger and load only my application debug information. How can I achieve that?*

A In fact if you want to start the debugger and load only the symbol of the application you have to specify a command file to be executed when the debugger is started rather than the name of the executable.

?? This is done using following command line in the Build Extras panel:

```
-c init.cmd -Prod=cisco.ini -instance=ESL
```

?? In the init.cmd file, insert the command to load symbols from the application:

```
LOAD bin\MyApplication.ABS SYMBOLSONLY
```

?? Note that you have to specify the name of the abs file there.

Q *I have problem to debug a Rom library and an application using the Elf/Dwarf format.*

A Our development tool chain currently does not allow loading debug information from a ROM library and an application at the same time.

?? In order to be able to debug inside of a ROM library, load the ROM library with "Symbols only".

?? Then you have to reload the application debug information when returning to the main application.

Q *How can I dump the content of the memory area in a file?*

A The command 'SAVE' allows you to save a memory area as an S record file. The S record file can then be re-loaded in the debugger later on. Take a look at the HI-WAVE engine manual for a description of the SAVE command.

Q *While debugging how can I check if some code part was optimized?*

A In the 'source' window click with the right mouse button. In the pop-up menu select the 'Marks' entry. Red marks appear in front of each statement, where you can set a breakpoint. Code has been generated for each instruction preceded with a Mark.

A In order to detect which variables or procedure have not been linked to the application (because of smart-linking), check the UNUSED OBJECT section in the map file.

Q *I want to debug an existing absolute file. I do not want to rebuild the project before debugging. How can I achieve that?*

A Take a look at the technical note TN203.

Q *I want to switch to another target interface in my project. How can I achieve this?*

A Take a look at the technical note TN205.

Q *I have a function, which is implemented in a header file and I am not able to debug it. What should I do in order to be able to debug it?*

A This is a known issue in HIWARE file format, but works fine in ELF DWARF file format. We recommend to switch to the ELF DWARF file format. Use option -F2 in compiler and assembler and change library paths.

Q *How can I avoid the application to be loaded in the target board when "Debug" button is hit in the IDE?*

A In order to avoid that the executable file is loaded to the target when you hit the Debug key:

?? Start CodeWarrior IDE

?? Open your project

?? Open the project Settings dialog.

?? Go to the "Build Extras" panel

?? In the External Debugger "Arguments" edit box remove "%targetFilePath -W".

?? Close the Project Setting dialog clicking OK.

This setting is specific to the project and will not apply to the other projects on the PC.

Q *I programmed the flash of my Star12, but suddenly I'm no more able to flash anything, including delivered samples and the communication is impossible or becoming very unstable.*

A These problems mostly occur on the star12 derivatives when you programmed the security byte located at address 0xFF0F. If you can communicate with your target board (depending on the target interface), scroll down in the debugger 'memory' window, at address 0xFF0F. If the value stored there is different than 0xFE, your chip is secured.

?? To unsecure the chip, simply click on the entry 'unsecure' in the target menu (i.e. SDI or ICD12...).

?? Note: on older versions, this entry is not present, you have to manually execute a command file delivered with the demo samples, here is the procedure:

?? In the debugger click on the menu 'component',

?? Choose 'open' and open a 'command' window

?? Click with the right mouse button in this window and select 'execute'.

?? Now browse for the erase_unsecure_dp256.cmd (i.e. the one in
{InstallDir}\Examples\HC12\ICD12 Target Interface\DP256\Fibo - Flash\cmd).

MMDS08/MMEVVS08

Q *I installed MCUEz and CodeWarrior on my PC and strangely, CodeWarrior seem to only take account of the mdsemcu.ini file of MCUEz (c:\mcuez\prog\), not of the one, which belongs to it (in the window directory i.e. C:\WinNT\). As an effect I can not see all the derivatives supported by CodeWarrior.*

A In fact, in your windows directory (i.e. C:\WinNT) MCUEz and CodeWarrior share the file mdselib.ini. This file contains an environment variable "Path" (defined in the Group [MDSEMUCU]), which specifies the path to the Mdsemcu.ini file. To fix your problem, simply edit the file (with CodeWarrior or any other ASCII text editor) and initialize this variable with the appropriate path

Mon08

Q *I am using the Mon08 target interface. I am using the PORTA in my application. The debugger Memory component does not display the modification of the PORTA during execution of the application.*

A Be careful, the HC08 monitor uses bit 0 in PORTA for synchronization purpose. Make sure not to modify bit0 in PORTA register. Always access this port using OR or AND logical operator rather than assignment.

Q *I am not able to debug my application, after I have placed some code at address 0xF0-0xFF.*

A The memory area 0xF0 to 0xFF is reserved for Monitor stacking. This memory should not be used by the application when debugging using the Mon08 monitor. In release mode, the whole RAM area is available for the application

Q *With the HC908AZ/AT/AS, I have the communication, I can load and debug in ram, but I'm not able to program all vectors.*

A It comes from the fact that this cpu family, starts by default in AS60 mode (which have less vectors). Our debugger respects the default setting. To fix this problem, in your project, simply put in the command file called startup.cmd the following commands:

```
wb 0xfe09 0x01
```

This will set the register MSCAND to AZ60 mode. Please refer to the mon08 manual for more details

Q *When I attempt to step over a function or out of a function, I get the message "No free resources to set the breakpoint. Delete or disable some other watchpoints or breakpoints"*

A When you are using features like "Step Over", "Step Out" or "Run To Cursor" the debugger internally set a breakpoint to the next instruction, which should be executed.

When the application is loaded into flash, the number of breakpoints and watchpoints you can set in the application is limited by the resource available on the CPU.

On a HC08 CPU, there is one hardware breakpoint register available. This means that at a given time you can set: only one breakpoints into flash memory.

If you have already defined one breakpoint, it will use the hardware breakpoint register. This means that the debugger is not able to step over a function located in flash any more.

HC12

Q *When I attempt to step over a function or out of a function, I get the message “No free resources to set the breakpoint. Delete or disable some other watchpoints or breakpoints”*

A When you are using features like "Step Over", "Step Out" or "Run To Cursor" the debugger internally set a breakpoint to the next instruction, which should be executed.

When the application is loaded into flash, the number of breakpoints and watchpoints you can set in the application is limited by the resource available on the CPU.

On a HC12 CPU, there are two hardware breakpoints registers available. This means that at a given time you can set:

?? Two breakpoints into flash memory or

?? One Watchpoint.

If you have already defined one watchpoint or two breakpoints, the two hardware breakpoint registers are used. This means that the debugger is not able to step over a function located in flash any more.

SDI

Q *I cannot communicate with my target board. External power is applied to the cable.*

A When you are working with a SDI target interface, and if you do power the SDI external (if you do not take power from the card), you have to make sure the card is powered first and the SDI afterward. If the SDI is powered before the card, connection will not work.

Note: In case you power the SDI externally, you also have to make sure you have moved the jumpers in the SDI cable in the appropriate position.

BDI HC12

Q *While debugging with my BDI, the routine in charge of the initialization of the EEPROM seems not to work or the datas vanished after performing a power-on, power-off.*

A On most of the HC12 derivatives, the EEPROM overlaps the registers, so Abatron, in their delivered init list, move the EEPROM at another location, and put RAM at the initial location. There are two solutions to fix this problem:

?? _Modify the init list

?? _Modify your application with the new address of the EEPROM (warning: when working in stand alone do not forget to modify the startup code and also remap the EEPROM to the appropriate address).

Simulator HC11

Q *How can I configure the simulator to debug banked application?*

A In order to allow the simulator to debug banked application, add following environment variables to the section [Environment Variables] in the project ini file.

[Environment Variables]

BANKED=ON

BANK_IO_ADR=0x58

BANK_BASE=0x8000

BANK_LIMIT=0xAFFF

?? BANKED=ON enables debugging of banked application

?? BANK_IO_ADR specifies the address of the register used as bank register.

?? BANK_BASE, BANK_LIMIT specifies the bank window.

Small Monitor (HC11)

Q *How can I configure the debugger to debug banked application?*

A In order to allow the debugger to debug banked application, add following environment variables to the section [Environment Variables] in the project ini file.

[Environment Variables]

BANKED=ON

BANK_IO_ADR=0x58

BANK_BASE=0x8000

BANK_LIMIT=0xAFFF

?? BANKED=ON enables debugging of banked application

?? BANK_IO_ADR specifies the address of the register used as bank register.

?? BANK_BASE, BANK_LIMIT specifies the bank window.

ICD HC16

Q *Does the HI-WAVE ICD target interface support my particular CPU16 MCU?*

A The debugger supports all MCUs from the CPU16 and CPU32. In order to be able to communicate with your hardware, you have to initialize your Chip select appropriately. The chip select must be initialized in the files startup.cmd, reset.cmd and forcebdm.cmd from your project.

Q *After starting up the ICD Debugger an error "Illegal breakpoint detected" appears. What is wrong?*

A The cable might be too long; the maximal length for unshielded cables is about 20 cm and depends on the electric noise in the environment.

Q *Problems in the communication, loosing communication*

A The cable might be too long; the maximal length for unshielded cables is about 20 cm and depends on the electric noise in the environment.

Simulator

Q *How to avoid the simulator announcing no access to memory*

A In the simulator change the memory configuration mode to "auto on access".

Q *How can the same memory configuration be loaded every time the simulator is started?*

A Save that memory configuration under default.mem ex: Simulator->Configure -> Save and type in "default.mem".

Q *How can I make sure my own debugger layout is loaded each time the debugger is started?*

A Once you have defined your debugger layout, select entry “Save Project” in the debugger File menu.