

Correspondence between MSL and EWL libraries in CodeWarrior products

1. Introduction

The purpose of this application note is to show how to find and use alternative library in terms of the MSL/EWL libraries perspective correctly. To find the right alternative libraries is as important as importing project created in CodeWarrior 2.x into CodeWarrior 10.x product.

The CodeWarrior build tools translate source code into object code and then organize the object code to create a program that is ready to execute. CodeWarrior build tools run on the host system to generate software that runs on the target system. The CodeWarrior compilers use the Embedded Warrior Library (EWL) for C to provide and extend the libraries documented in the ISO/IEC standards for C programming. The Main Standard Libraries (MSL) is complete and configurable set of standard libraries. The Embedded Warrior Library (EWL) is the next generation of MSL libraries. The EWL library uses the definitions of preprocessor macros to specify the capabilities of the platform on which EWL runs. The EWL libraries provide a prefix file for each build target that EWL runs on. The prefix file is header file that contains a list of macro definitions.

Contents

1. Introduction.....	1
2. Main Standard Libraries Description	2
3. Embedded Warrior Library Description	3
4. Runtime Libraries Description.....	3
5. Finding the Right Alternative Files from the MSL/EWL Libraries Perspective	4

2. Main Standard Libraries Description

Main Standard Libraries (MSL) is a complete, configurable set of C and C++ standard libraries. These libraries also include MSL Extras, which extends the standard library and adds compatibility with common UNIX libraries. To use the MSL library, there must be also used a runtime library. If a user application uses features in the MSL libraries, the user must choose a configuration that matches your software and hardware requirements. The file names of the configurations of MSL libraries follow a naming convention to describe each library's capabilities and features. There is definition of MSL libraries configurations in Table 1.

Table 1. MSL Libraries Configurations Definition

MSL Suffix Name	MSL Suffix Name Equivalent:
Fdlibm	High-level math functions, including as the trigonometric functions
MSL_C	C standard library
MSL_C++	C++ standard library
MSL_EC++	Embedded C++ standard library
MSL_SUPP_TRK	CodeWarrior TRK support
MSL_SUPP_UART	UART for serial communications
.bare	Boards with no operating system
PPCEABI	Conforms to the PowerPC Embedded Application Binary Interface (EABI) standard
SZ	Optimized for size
SP	Single Precision Floating Point only
A	AltiVec support
C	Code compression
E	e500 and e200z (formerly Zen) targets
E2	e500v2 targets, with double-precision floating-point operations
H	Hardware floating-point operations
HC	Hardware floating-point operations and code compression
S	Software emulation of floating-point operations
N	No floating-point support
NC	No floating-point support, but with code compression
LE	Little-endian mode
UC	Function parameters declared char are treated as if they were declared unsigned char
V	Uses VLE instructions SPFP/SPE floating point operations in software routines. Use only with processors that have an e200z (formerly Zen) core
VS	Uses VLE instructions, single-precision floating point operations using native processor instructions, and double-precision floating point operations using software routines
SC	Function parameters declared char are treated as if they were declared signed char

3. Embedded Warrior Library Description

Embedded Warrior Library (EWL) is the next generation of MSL libraries. With this release, EWL will be an alternative library. The sources are based on MSL and are MISRA compliant. Existing standard prefix file name, library (archive) names have been modified. Some of the legacy libraries have been deprecated and a greater number of processor core specific libraries are introduced. EWL libraries are not supported by all products. All of your existing projects and makefile access paths will not use the EWL unless you specifically select it. The each name of EWL library has prefix, core and flags pieces. There is list the details of each piece of MSL libraries name below.

The prefix is one of the following:

- *libm_* - the library has math features
- *librt_* - the library has runtime features
- *libc_* - the library has reduced code size C features
- *libc99_* - the library has faster and increased C99 conformant C features
- *libstdc++_* - the library has latest C++ features
- *libc++_* - the library has reduced code size C++ features

The core starts with the processor family (like e200 or e500) and optionally ends with the core name (such as z750).

The current flags are VLE, Soft (software floating point) and SPFP_Only (math library only has single precision sources and source file doubles are treated as if they are single precision). SPFP_Only is only used with e200 and e500 which have single precision floating point instructions but there are not double precision instructions there.

4. Runtime Libraries Description

Runtime library provides low-level functions that support high-level C and C++ language features, such as memory management and file system access. Conceptually, a runtime library acts as an interface between a target system's hardware or operating system and the CodeWarrior C or C++ runtime environment. Substrings embedded in the name of a runtime library indicate the type of support the library provides. Use these substrings to pick the runtime library appropriate for the user project. There is definition of meaning of each library's file name substring in Table 2.

Table 2. Runtime Libraries File Name Substrings Definition

MSL Suffix Name	MSL Suffix Name Equivalent
Runtime	The library is a C language library.
Run_EC++	The library is an embedded C++ library.
PPCEABI	The library conforms to the PowerPC Embedded Application Binary Interface (EABI) standard.
A	The library provides AltiVec support.
E	The library is for e500 and e200z (formerly, Zen) targets.

E.fast	The library is for e500 and e200z (formerly, Zen) targets. Further, this library's floating-point operations are faster than those of a .E library, but they do not strictly conform to the IEEE floating-point standard.
E2	The library is for e500v2 targets and supports double-precision floating point operations.
H	The library supports hardware floating-point operations.
HC	The library supports hardware floating-point operations and code compression.
S	The library provides software emulation of floating-point operations.
SP	Single Precision Floating Point only.
N	The library provides no floating-point support.
NC	The library provides no floating-point support, but supports code compression.
LE	The library is for a processor running in little-endian mode.
UC	The library was built with the Use Unsigned Chars option of the C++ Language target settings panel enabled. As a result, all library function parameters declared char are treated as if they were declared unsigned char. Use a UC library in build targets for which the Use Unsigned Chars option is enabled. Use a non-UC library in build targets for which this option is disabled. If the option used by the build target is different from the option used to generate the build target's runtime library, the linker issue a warning.
V	Contain VLE instructions. Perform single-precision floating point operations using the core's SPE auxiliary processing unit (APU) or double-precision floating using software routines.
VS	Contain VLE instructions. Perform all floating-point operations using software routines Use only with processors that have an e200z (formerly Zen) core

5. Finding the Right Alternative Files from the MSL/EWL Libraries Perspective

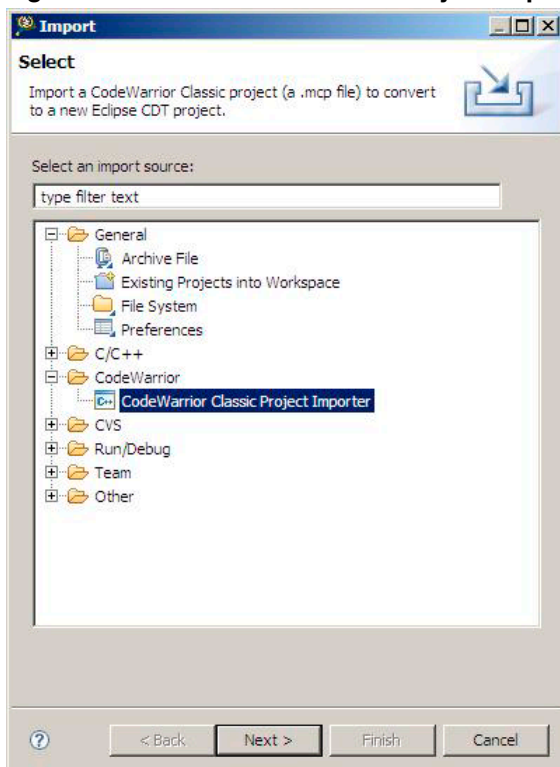
The prefix name used in EWL libraries also differs from that of the MSL prefix name. For example, there is **ansi_prefix.PPCEABI.bare.h** prefix name in MSL libraries referred to as **ansi_prefix.PA_EABI.bare.h** prefix name in EWL libraries.

5.1. Import an Existing Project

To import a project from CodeWarrior 2.x to the CodeWarrior 10.x (Eclipse-based product), perform the following steps:

1. Select item **File > Import** from main menu of the CodeWarrior 10.x product and open **CodeWarrior > CodeWarrior Classic Project Importer** window from the list of items, as shown in Figure 1.

Figure 1. CodeWarrior Classic Project Importer



2. Browse the **.mcp** project's file type of an existing project and finish the project importer wizard. When import process of the current project is done the importer will convert all of used MSL libraries to equivalent EWL libraries automatically.
3. Check the suffix names of files in a target project.

Sometimes there may not be the libraries converted properly. It is necessary to check the suffix name in MSL libraries and then find equivalent in EWL libraries from the table below. If there is used any prefix files something like **ansi_prefix.PPCEABI.bare.h** header file in the project then is necessary to make sure to change equivalent EWL library prefix files.

There is also possible to find more than one library for the same suffix name. There must be library selected accordingly for which processor is used in this case. The processor's type is a part of library's name, e.g. for following MSL libraries:

Runtime.PPCEABI.V.UC.a
MSL_C.PPCEABI.bare.V.SP.UC.a

There exists equivalent in EWL libraries like following:

librt_E200z335_VLE.a
libc_E200z335_VLE_SPFP_Only.a

5.2. Table of Equivalent MSL/EWL Libraries Names

Use the following table to find the equivalent file name of MSL/EWL libraries.

Table 3. Equivalent File Names of EWL and MSL libraries

MSL Suffix Name Equivalent:	EWL Library Core and Flag Name:
PPCEABI.VS.UC	E200z0_VLE_Soft
PPCEABI.VS.UC	E200z150_VLE_Soft
PPCEABI.V.UC	E200z335_VLE
PPCEABI.V.SP.UC	E200z335_VLE_SPFP_Only
PPCEABI.V.UC	E200z336_VLE
PPCEABI.V.SP.UC	E200z336_VLE_SPFP_Only
PPCEABI.V.UC	E200z446_VLE
PPCEABI.V.SP.UC	E200z446_VLE_SPFP_Only
PPCEABI.V.UC	E200z448_VLE
PPCEABI.V.SP.UC	E200z448_VLE_SPFP_Only
PPCEABI.E.UC	E200z650
PPCEABI.E.SP.UC	E200z650_SPFP_Only
PPCEABI.V.UC	E200z650_VLE
PPCEABI.V.SP.UC	E200z650_VLE_SPFP_Only
PPCEABI.E.UC	E200z652
PPCEABI.E.SP.UC	E200z652_SPFP_Only
PPCEABI.V.UC	E200z652_VLE
PPCEABI.V.SP.UC	E200z652_VLE_SPFP_Only
PPCEABI.V.UC	E200z750_VLE
PPCEABI.V.SP.UC	E200z750_VLE_SPFP_Only
PPCEABI.V.UC	E200z760_VLE
PPCEABI.V.SP.UC	E200z760_VLE_SPFP_Only
PPCEABI.H.UC	E300c1
PPCEABI.H.UC	E300c2
PPCEABI.H.UC	E300c3
PPCEABI.H.UC	E300c4
PPCEABI.E.UC	E500V1
PPCEABI.E.SP.UC	E500V1_SPFP_Only
PPCEABI.E2.UC	E500V2
PPCEABI.A.UC	E600

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale, the Freescale logo, CodeWarrior and ColdFire are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Flexis and Processor Expert are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners

© Freescale Semiconductor, Inc. 2013. All rights reserved.

Document Number: AN4678

28 February 2013

