



Quick Start Guide

BeeKit™ Wireless Connectivity Toolkit



Contents

About This Guide	2
Audience	2
Revision History	2
Conventions	3
Definitions, Acronyms, and Abbreviations	3
BeeKit Introduction	4
System Requirements	4
Required Hardware and Software	5
Installation	6
Considerations for First Time Installations	8
Compiler/Debugger IDE Considerations	8
Evaluation Board Driver Considerations	8
Creating a Demonstration Solution	9
Customizing Project Parameters	18
Exporting and Loading the Solution	20
Updating CodeWarrior Software	21
Importing BeeKit Solutions and Projects to CodeWarrior 10.1	23
Importing BeeKit Solutions to IAR Embedded Workbench	26
Exporting and Loading BeeKit Solutions to IDEs from the BeeKit GUI	27
Application Development	29

About This Guide

This guide describes how to install and run Freescale’s BeeKit[®] Wireless Connectivity Toolkit. This guide describes system requirements, the installation procedure and provides users with a basic introduction about how to use BeeKit for 802.15.4 MAC, SMAC, SynkroRF, ZigBee[®] or RF4CE application development.

This guide also provides a simplified, step-by-step process that shows how to open and load a demonstration project using either Freescale CodeWarrior[®] for HCS08 based projects or IAR Embedded Workbench[®] for ARM7[®] MC1322x based projects without further modification. This allows users to more quickly evaluate Freescale’s ZigBee technology.

The example used in this guide allows users to select either the ARM7 MAC Codebase or the HCS08 802.15.4 MAC Codebase.

For more information on ZigBee, visit www.freescale.com/zigbee.

Audience

This document is intended for ZigBee, BeeStack Consumer (RF4CE), IEEE[®] 802.15.4 MAC, SMAC and SynkroRF custom software and application developers.

Revision History

Current document revision number: 2.4

The following table summarizes revisions to this document since the previous release (Rev 2.3).

Revision History

Location	Revision
Entire document	Updated for CodeWarrior 10 and MC1323x silicon.

Conventions

This document uses the following notational conventions:

- `Courier monospaced type` indicate commands, command parameters, code examples, expressions, datatypes, and directives.
- *Italic type* indicates replaceable command parameters.
- All source code examples are in C.

Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

BDM Debugger	A debugger that uses the BDM interface. For example, the P&E BDM Multilink debugger
BDM	Background Debug Module
IDE	Integrated Development Environment
JTAG Debugger	A debugger that uses the JTAG interface. For example, the J-Link JTAG Debugger
SMAC	Simple Medium Access Control
Solution	A Solution is a Project container. It is conceptually a grouping for related Projects or wireless applications. For example, while using BeeKit, a user will be working on a Solution, even if the Solution contains just a single Project.
Project	A Project consists of assembled Codebase components (or building blocks) which, in the end, are compiled into a single wireless application.
Workspace	A Workspace is all the files needed by the CodeWarrior IDE to compile an application. The Workspace is where users add their own files if using the CodeWarrior IDE.
Codebase	A Codebase contains all the information required for BeeKit to interact with the user and allow them to create their Project. Freescale provides different Codebases for its BeeStack (ZigBee), IEEE 802.15.4 MAC and SMAC products (both HCS08 and ARM7).

BeeKit Introduction

The Freescale BeeKit Wireless Connectivity Toolkit (BeeKit) is a comprehensive collection of Codebases consisting of wireless networking libraries, application templates and sample applications for Freescale's family of IEEE 802.15.4 and ZigBee transceivers. The BeeKit Graphical User Interface (BeeKit GUI), part of BeeKit, allows users to create, modify, and update various wireless networking implementations.

NOTE

Codebases are embedded software and are not described in this guide.

Updates for the BeeKit GUI and Codebases are located at www.freescale.com/zigbee. Freescale recommends users visit this web site often to ensure that they have the latest BeeKit and Codebase releases.

System Requirements

BeeKit runs on the following Freescale development kits.

- 1323x-DSK Developer Starter Kit
- 1323x-NSK Network Starter Kit
- 1320x-QE128-DSK Developer Starter Kit
- 1322xDSK Developer Starter Kit
- 1322xNSK Network Starter Kit
- 1322xEVK ZigBee Evaluation Kit
- 13226Pro Kit
- 1321xDSK Developer Starter Kit
- 1321xNSK Network Starter Kit
- 1321xEVK ZigBee Evaluation Kit
- 1321xCSK Consumer Starter Kit
- 13192DSK Developer Starter Kit
- 13192EVB Evaluation Board

- 13192EVK ZigBee Evaluation Kit
- 1320xRFC RF Daughter Card
- 13192RFC RF Daughter Card

Required Hardware and Software

- The BeeKit installer package and documentation are available from www.freescale.com/802154
- For MC1319x, MC1320x, MC1321x, and MC1323x HCS08 based solutions:
CodeWarrior Development Studio for Freescale HCS08 Microcontrollers, Version 10.1 updated with all available Service Packs (Full License)
- For MC1319x, MC1320x, MC1321x, and MC1323x HCS08 based boards:
USB HCS08/HCS12 Multilink BDM from P&E Microcomputer Systems, Inc
- For MC1322x ARM7 based solutions:
IAR Embedded Workbench for ARM (Requires 256K Edition for use with IEEE 802.15.4 MAC, SynkroRF, BeeStack and BeeStack Consumer codebases). Codebases supporting the MC13226 require version 5.50 of IAR Embedded Workbench. Codebases supporting the MC13224 require version 5.20 of IAR Embedded Workbench
- For MC1322x ARM7 based boards:
J-Link JTAG Debugger
- Windows 2000[®], Service Pack 3 or later, Windows XP[®], Service Pack 1 or later, Windows Vista[®], or Windows 7[®]
- Windows administrator user privileges
- Internet access for software downloads
- Microsoft .NET Framework 2.0 Runtime available from www.microsoft.com
- Microsoft .NET Framework 3.5 SP1 and Windows Media Player 9 or later are required for running the MC1323x HECA Demo

Installation

1. Launch the BeeKit installer `BeeKitSetup.exe` downloaded from the Freescale web site or from the BeeKit CD-ROM and click the “Next” button to start the installation process.



Figure 1. Installer Window

2. To complete the installation, follow the instructions as they appear on the screen.

3. After installation is complete, from Windows, click on Start -> Programs -> Freescale BeeKit -> Freescale BeeKit. The BeeKit main window appears as shown in Figure 2.

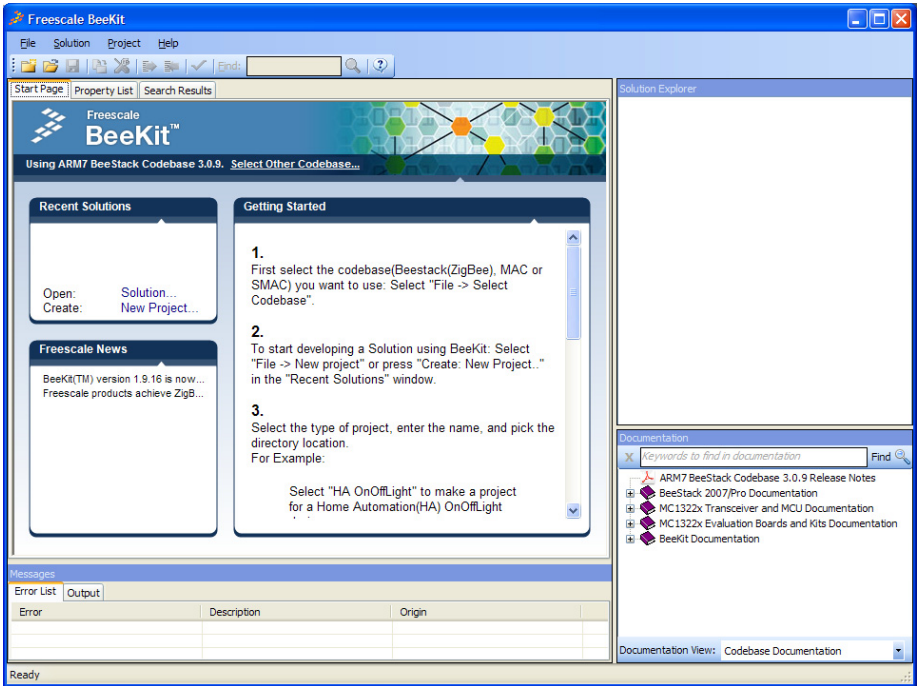


Figure 2. BeeKit Main Window



Considerations for First Time Installations

Compiler/Debugger IDE Considerations

Depending on the platform chosen for development, either the Freescale CodeWarrior IDE or the IAR Embedded Workbench IDE must be installed before running BeeKit for the first time. The IDEs are available on separate media with a Freescale Evaluation Kit or can be downloaded or purchased online at <http://www.freescale.com/codewarrior> or <http://www.iar.com>.

Evaluation Board Driver Considerations

BeeKit must be installed before plugging any development boards into the system. When users first plug a board into the system, they may be prompted to install drivers. If this occurs, do not allow Windows to automatically search for and install the drivers. Instead, select manual installation and steer Windows to the following directory:

```
C:\Program Files\Freescale\Drivers
```

If installing the BeeKit software package to another drive or directory, indicate the Drivers directory created by the installer in the custom location where BeeKit was installed.

Complete driver installation by following the instructions as they appear on screen.

After BeeKit and the hardware drivers are installed, Windows can then be allowed to automatically search for and install drivers when other Freescale boards are connected to the system.

Creating a Demonstration Solution

Demonstration applications are fully functional, embedded applications that can be downloaded and tested on Freescale wireless devices. No programming expertise is required.

The BeeKit Codebases contain several pre-configured demonstration applications that are loaded on various demonstration boards contained in the evaluation kits. This example uses the MyStarNetwork Demo (Coordinator) and MyStarNetwork Demo (End Device) application templates from either the HCS08 or the ARM7 IEEE 802.15.4 MAC Codebase. To create this demonstration application, perform the following steps.

1. Select the appropriate Codebase (MAC). The example used in this guide allows users to select either the ARM7 MAC Codebase or the HCS08 MAC Codebase. From the BeeKit main window (Figure 2), click on the “Select Other Codebase...” button or from the menu bar click on File -> Select Codebase. A Codebase selection window appears as shown in Figure 3. Select the appropriate MAC Codebase version and click the “Set Active” button.

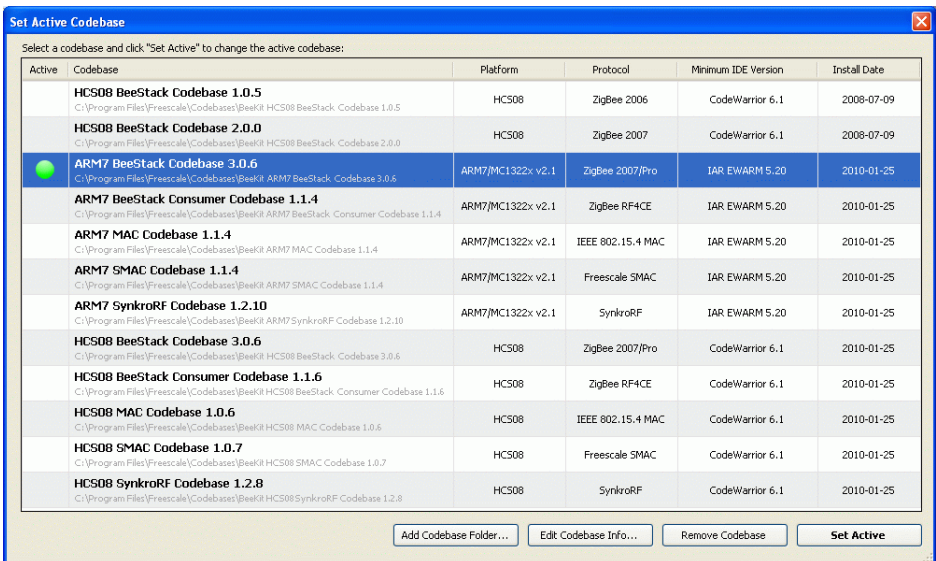


Figure 3. Codebase Selection

2. From the BeeKit main window (Figure 2), to open a new Project, users have three options:

- Click on the New Project icon in the toolbar
- Select File -> New Project from the menu bar
- In the “Recent Solutions” display area, click on “New Project”

The New Project window appears as shown in Figure 4.

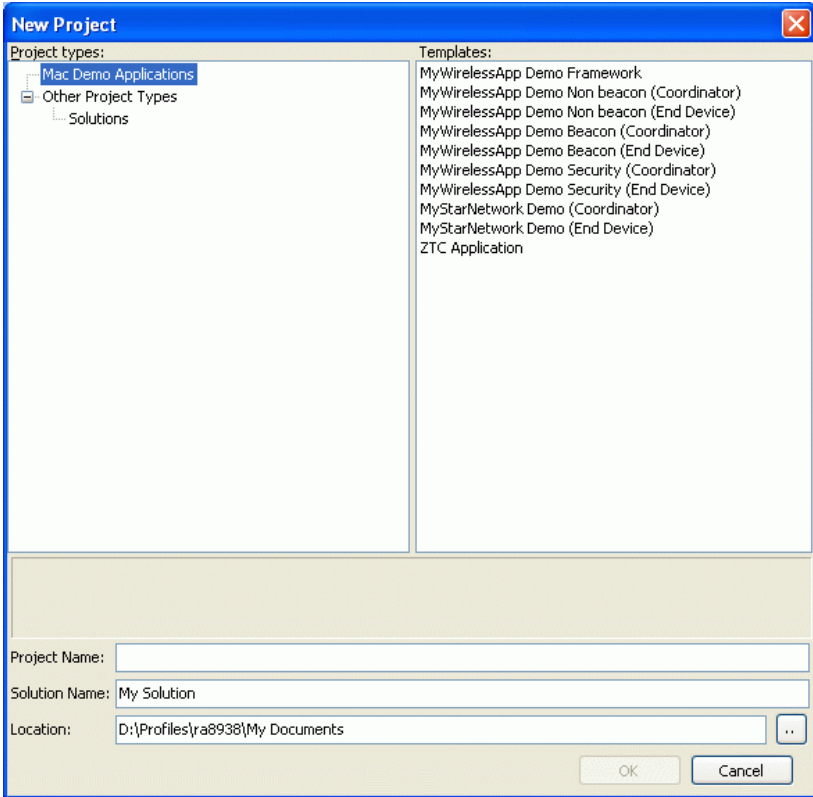


Figure 4. New Project Window

3. For this example, the “MAC Demo Applications” option is highlighted in the New Project Window, Project types area.

4. Select “MyStarNetwork Demo (Coordinator)” from the New Project window Templates area. This creates a Project for implementing the coordinator side of a star network as shown in [Figure 5](#).

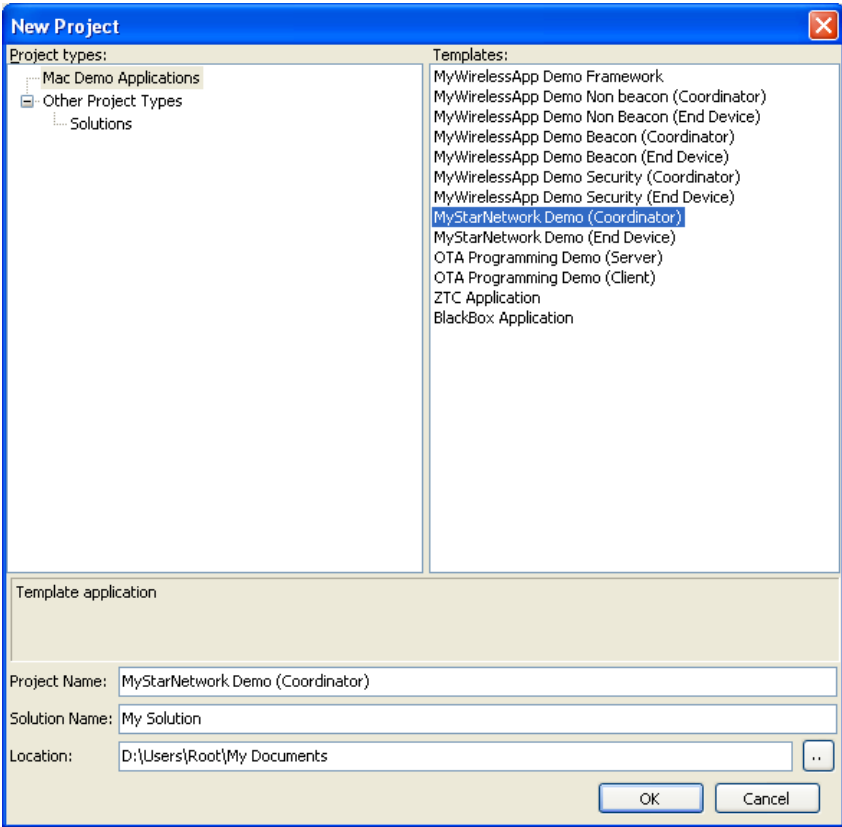


Figure 5. Directory Location and Project Names Window

The Project Name field is automatically filled in when users select a Project from the templates area of this window. However, the Project Name and Solution Name fields can be changed to more closely match application functionality.

5. In the Solution Name field, enter an appropriate name for this Solution. The default name is “My Solution”. In this example, the Solution Name field was changed to “MyStarNetwork Solution”.

6. In the Location field, enter an appropriate directory location for this Project. The default location is the current Windows user directory “My Documents”. In this example, the location was changed to the “C:\802.15.4 MAC Projects” directory.
7. Clicking the OK button creates the Solution and the BeeKit Project Configuration window appears as shown in [Figure 6](#).

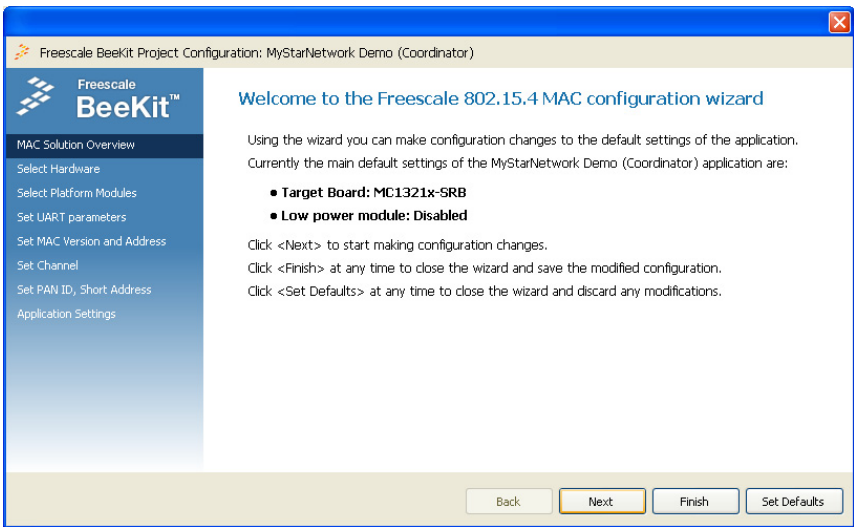


Figure 6. BeeKit Project Configuration Window

The BeeKit Project Configuration window displays various options on the window’s left side. Because these are protocol specific options, the SMAC, IEEE 802.15.4 MAC and Freescale BeeStack protocols each have independent BeeKit Project Configuration window display options.

The Overview option area displays a summary of user platform settings. For example, [Figure 6](#) shows the following Project settings for the MyStarNetwork Demo (Coordinator):

- MC13122x Network Node (the target board)
- Low power module: Disabled

Starting at this window, users can select the hardware and platform settings, basic configuration options and application specific parameters, etc. depending on the SMAC, IEEE 802.15.4 MAC or BeeStack protocol being used.

8. Click on the “Select Hardware” option, or the “Next” button and one of two windows will appear as follows:

- If users have selected a Codebase that supports both the MC13224 and MC13226, the MC1322x Development Board hardware version selection window appears first as shown in [Figure 7](#). This window allows users to select either the MC13224 or MC13226 depending on which development board is being used. Select the MC13224.
- If users have selected a Codebase that does not support both the MC13224 and MC13226, the ARM7 Codebase hardware target board selection window appears first as shown in [Figure 8](#).

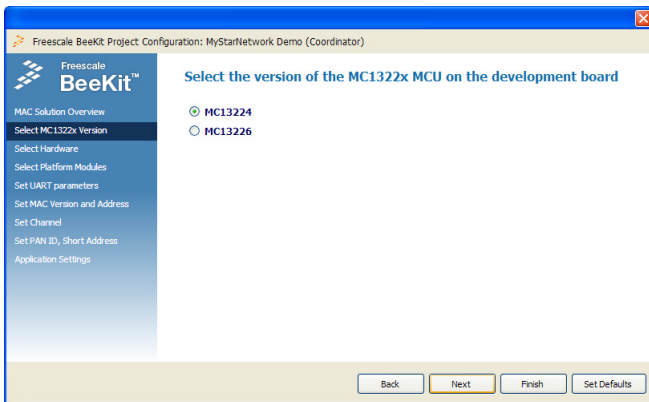


Figure 7. MC13224/MC13226 Selection Window

[Figure 8](#) shows the ARM7 Codebase hardware target board selection window. Select the MC1322x Network Node.

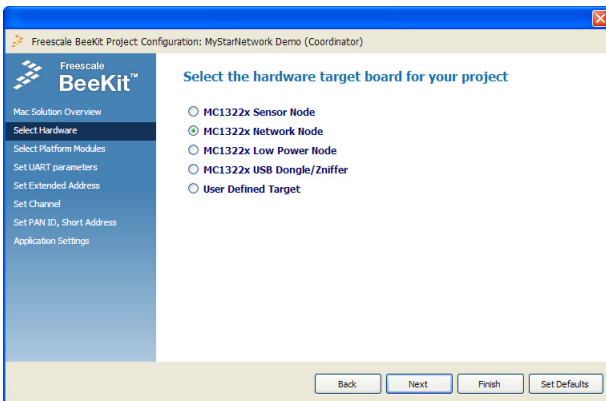


Figure 8. ARM7 Hardware Target Board Selection Window

Figure 9 shows the HCS08 Codebase hardware target board selection window.

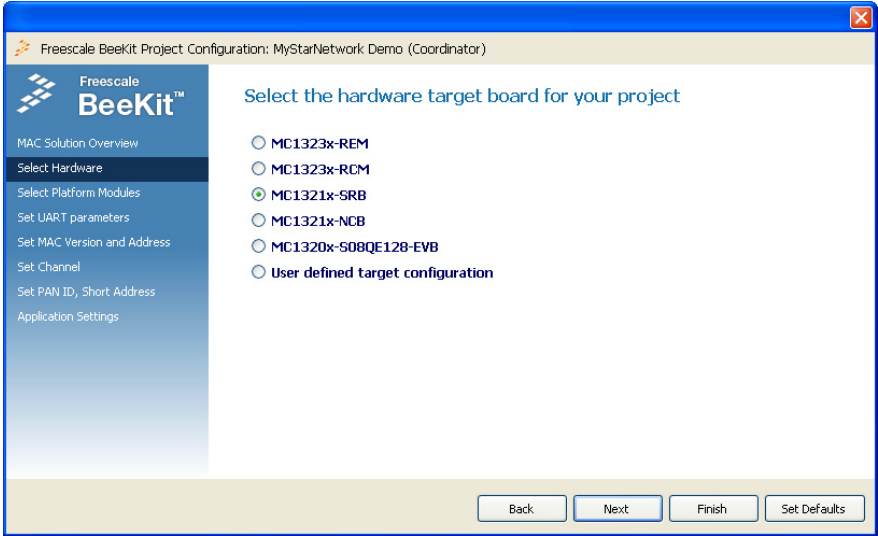


Figure 9. HCS08 Hardware Target Board Selection Window

- 9. Select the type of board to use and click the “Next” button.
- 10. The platform modules selection window appears as shown in Figure 10.

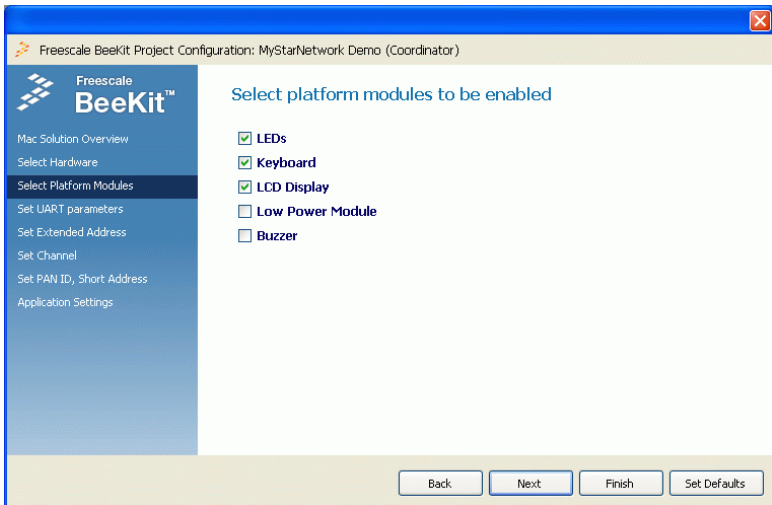


Figure 10. Platform Modules Selection Window

11. Do not change any of the default options and click the “Next” button.
12. The UART communications settings window appears as shown in [Figure 11](#).

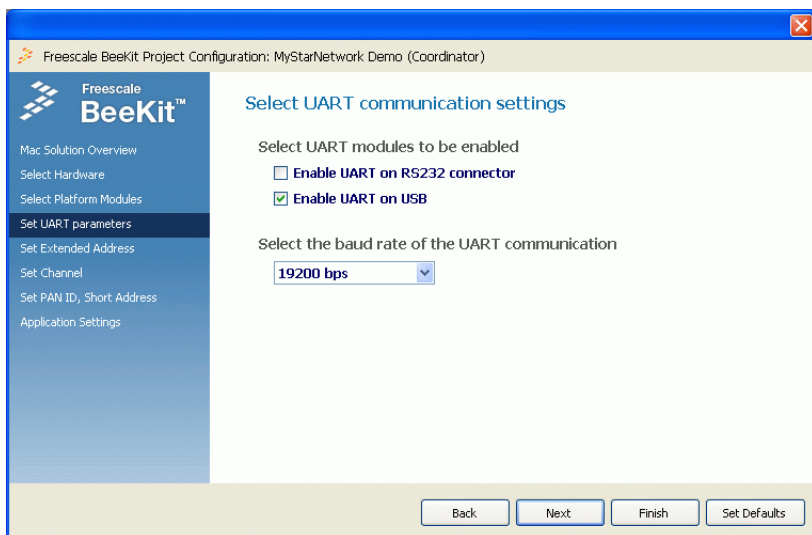


Figure 11. UART Communications Settings Window

13. Do not change any of the default options and click the “Next” button. The IEEE 802.15.4 Protocol Version and Extended (MAC) Address window appears as shown in [Figure 12](#).

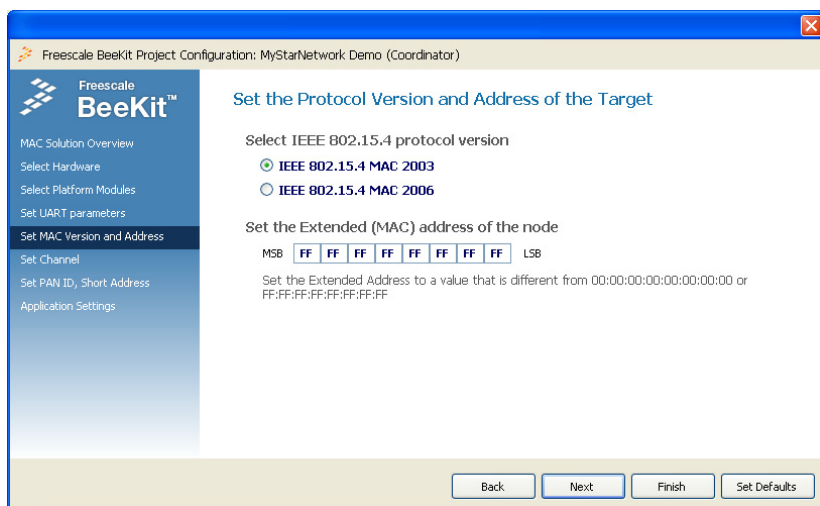


Figure 12. Extended (MAC) Address Window

14. This is the target board's MAC address. Any board that communicates with another board must have a unique Extended (MAC) Address. Enter a different address for each board in the input fields of this window. The original board MAC address printed on a label on the board's backside may be used. After entering an address, click the "Next" button. The RF channel selection window appears as shown in Figure 13.

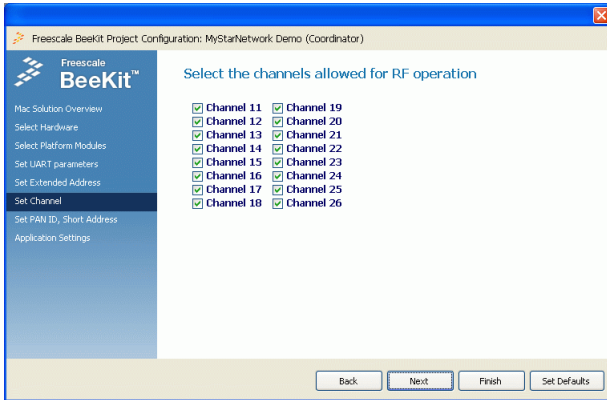


Figure 13. RF Channel Selection Window

15. Do not change any of the default settings because all channels are used for communication. Click the "Next" button.

16. The PAN ID and Node Short Address window appears as shown in Figure 14. The PAN ID must be the same for all devices that need to communicate with each other. Enter the settings as needed and click the "Next" button.

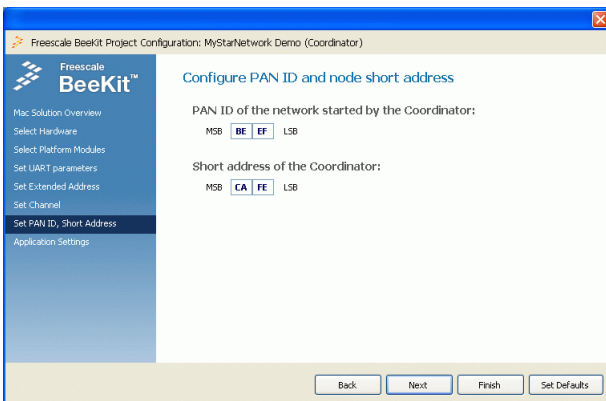


Figure 14. PAN ID and Short Address Window

17. The Application Specific settings window appears as shown in [Figure 15](#). This window is different for the various application templates in the 802.15.4 MAC Codebase. Customize the settings as needed and click the “Finish” button.

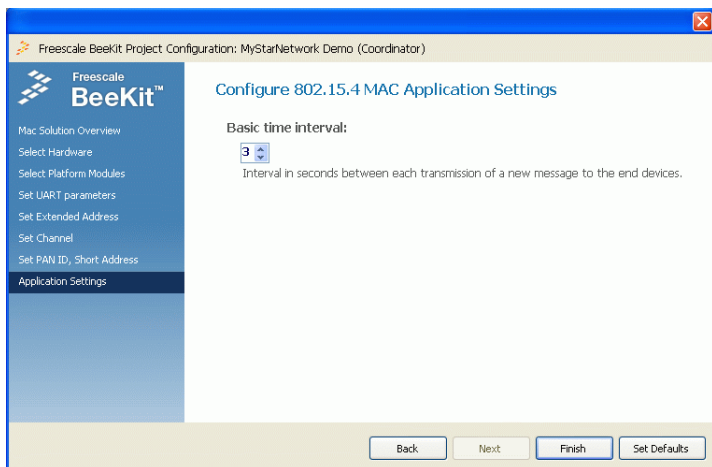


Figure 15. Application Specific Window

18. The BeeKit main window appears again as shown in [Figure 16](#). The Solution Explorer area displays the “MyStarNetwork Solution” containing the “MyStarNetwork Demo (Coordinator)” Project and its software components.

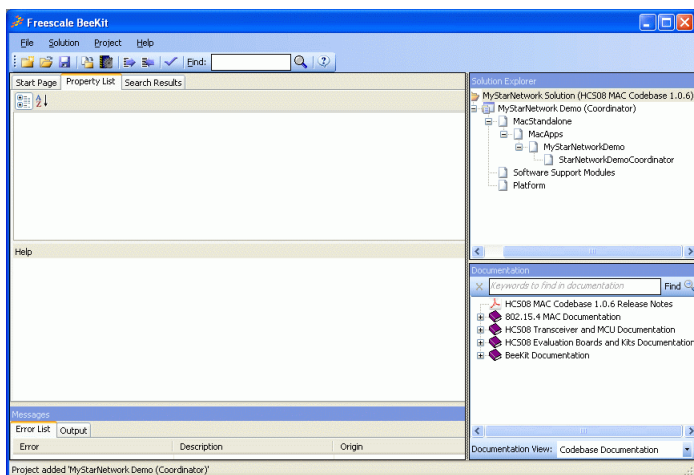


Figure 16. BeeKit Main Window with New Project Added

Customizing Project Parameters

Once an application Project is created, customize the application parameters by navigating to the different project components in the Solution Explorer panel and adjusting properties in the Property List tab for the selected component. Figure 17 shows the Property List for the StarNetworkDemoCoordinator project component.

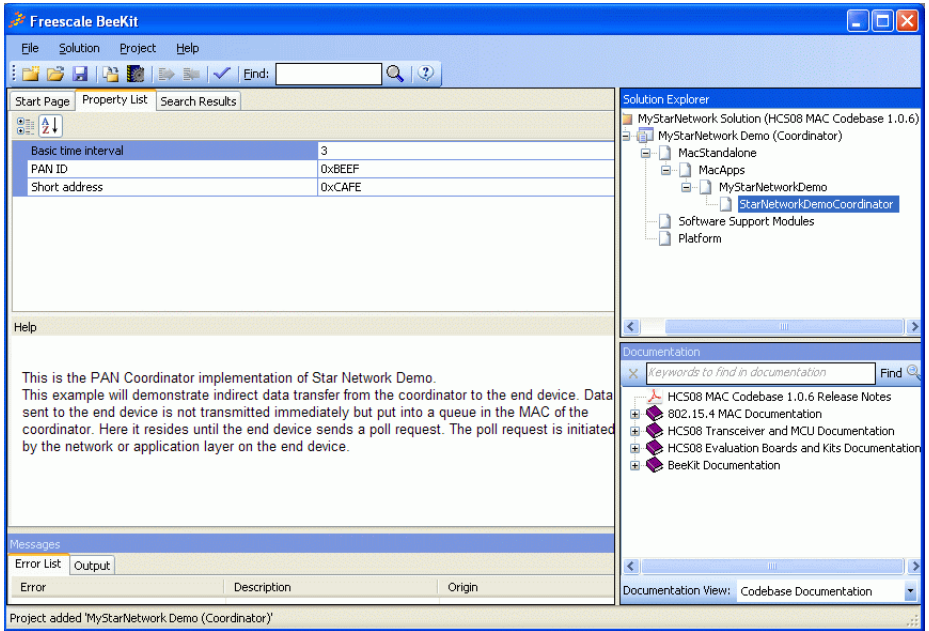


Figure 17. Property List for a Selected Project Component

Another way to quickly display and modify the configuration properties available in the BeeKit project components for a solution is to search for a specific property name or string. For example, perform the following steps to perform a search for the Target Hardware property which indicates which type of development board the solution is configured for.

1. In the Solution Explorer window select the Solution name to perform a solution wide search or just a specific Project name to limit the scope of the search to the components of the selected project.

2. In the BeeKit tool bar, enter the search string in the Find text box (For example, “Target Hardware”, as shown in Figure 18).

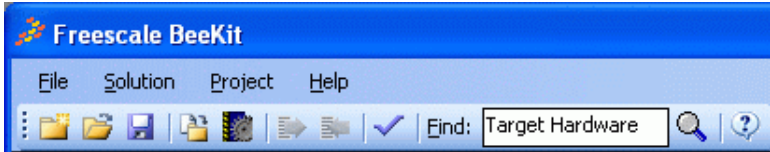


Figure 18. Entering a Search String

4. Press Enter or click the magnifying glass icon.
5. BeeKit displays the property or properties that have a name or that correspond to a compiler define which matches the searched string as shown in Figure 19.

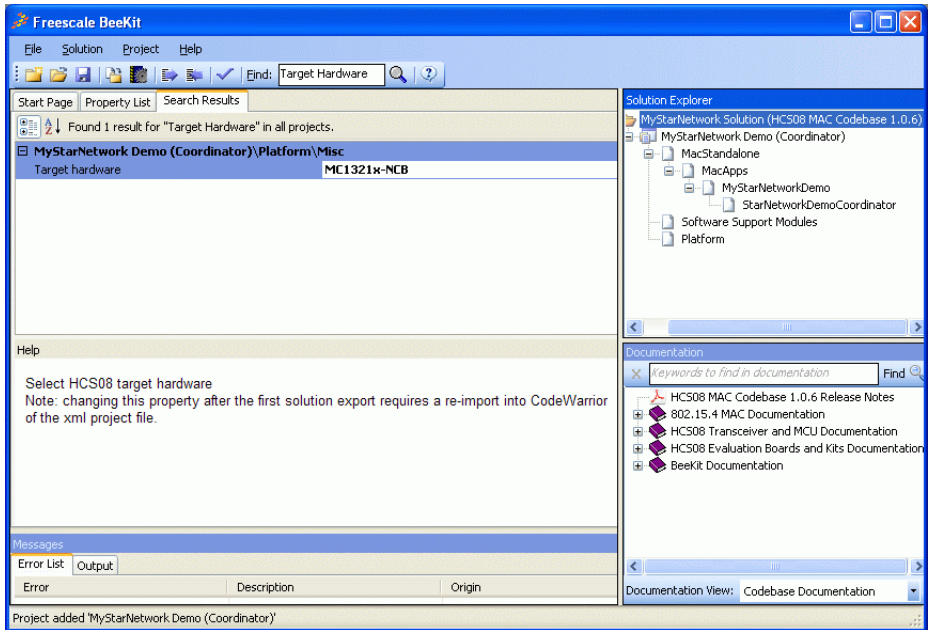


Figure 19. BeeKit Displaying Results for a Solution Wide Search

Exporting and Loading the Solution

Once a Solution is created, related Projects are added to that Solution and software components are defined and customized in the Projects. Users must export the Solution for deployment in either the Freescale CodeWarrior for HCS08 based Codebases or the IAR Embedded Workbench for ARM7 for MC1322x based Codebases.

BeeKit automatically displays export options and creates Project configuration files for one of the specific deployment IDEs based on the Codebase type. Exporting and loading the Solution to the corresponding IDE is described in the following sections.

To export the Solution and create the IDE project folders, perform the following steps:

1. From the menu bar of the BeeKit main window (Figure 16), click on: Solution -> Export Solution...

The Export Solution window appears as shown in Figure 20.

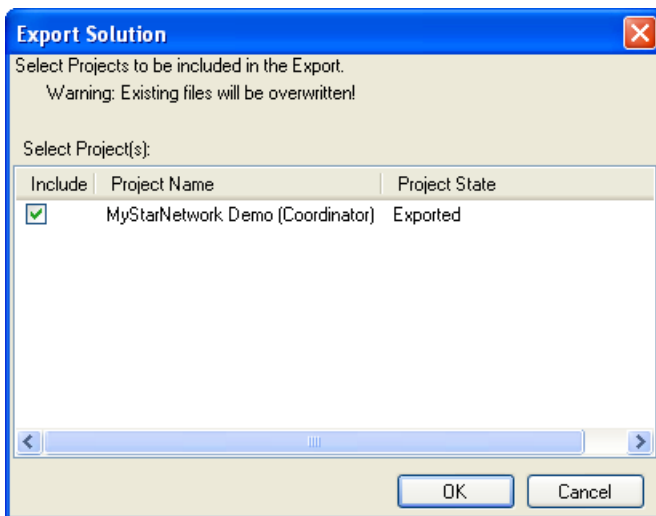


Figure 20. Export Solution Window

2. Ensure that the Project to be exported is checked so that it is included and click the “OK” button. Various messages appear in the message display area of the BeeKit main window as the Solution is exported.

3. Take note of the location where the solution was exported as shown in the last line of the export log as shown in [Figure 21](#).

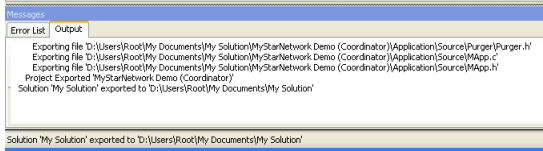


Figure 21. Export Solution Log

Updating the CodeWarrior Software

WARNING

CodeWarrior 10.1 (Build ID: 110204) IDE must be updated for the MC1323x platform and BeeKit compatibility.

To update the CodeWarrior IDE and all required plug-ins to their latest versions, perform the following steps:

1. Ensure a stable, broadband internet connection exists on the PC where the update will occur.
2. From the CodeWarrior 10.1 menu bar choose Help -> Install New Software...
3. From the Install window in the “Work with” combo box, use the arrow on the right side of the box to expand the Software repository list and select “FSL MCU Eclipse Update Site” as shown in [Figure 22](#).

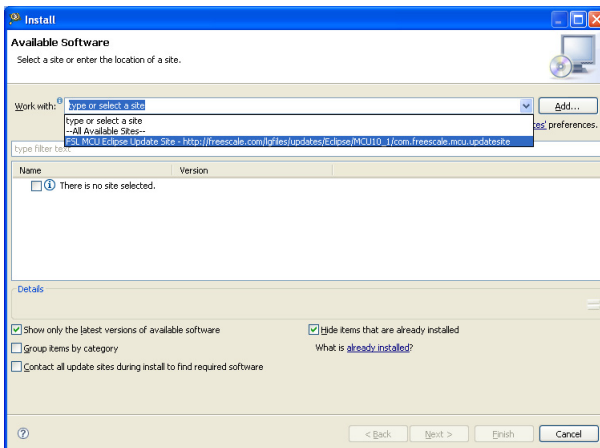


Figure 22. Update Install Window Repository Selection

4. At the bottom of the Install window (Figure 22), ensure that the “Hide items that are already installed” check box is selected and the “Contact all update sites during install to find updated software” is not selected.
5. A list appears that shows the available updates as shown in Figure 23.

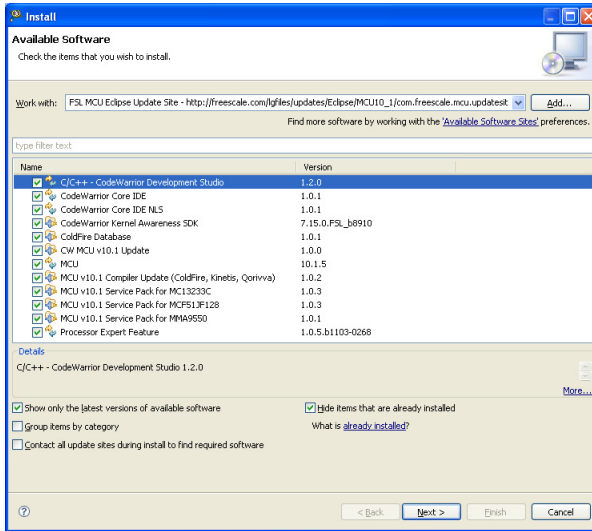


Figure 23. List of Available CodeWarrior Updates

6. Select the check boxes for all the updates displayed in the list.
7. Click Next to move to the “Install Details” window as shown in Figure 24.

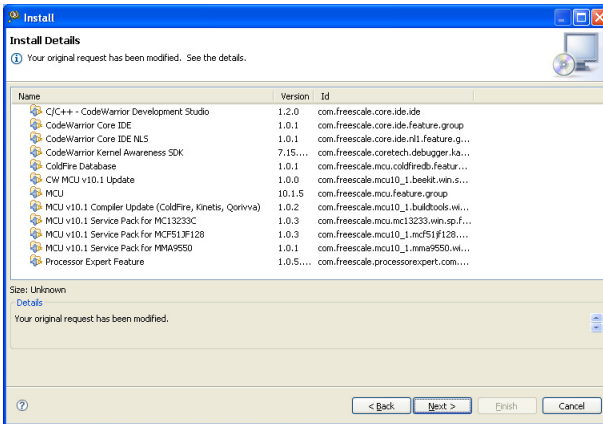


Figure 24. Install Updates Window

8. Click Next to move to the Review Licenses window. Users must review and accept the license agreements to proceed by selecting the corresponding options.
9. Click Finish to start the update process.
10. During the update process, user interaction may be required to confirm installation of some update packets or to restart the IDE.
11. If connection errors occur during the update process, restart the update process beginning at Step 1.

Importing BeeKit Solutions and Projects to CodeWarrior 10.1

To import a BeeKit solution to CodeWarrior 10.1, perform the following steps.

1. Open the CodeWarrior IDE and select a workspace.
2. Close the CodeWarrior Welcome Screen panel if it is still visible.
3. From the CodeWarrior Menu Bar choose File -> Import...
4. In the Import window expand the General section and select “Existing Projects into Workspace” as shown in [Figure 25](#).

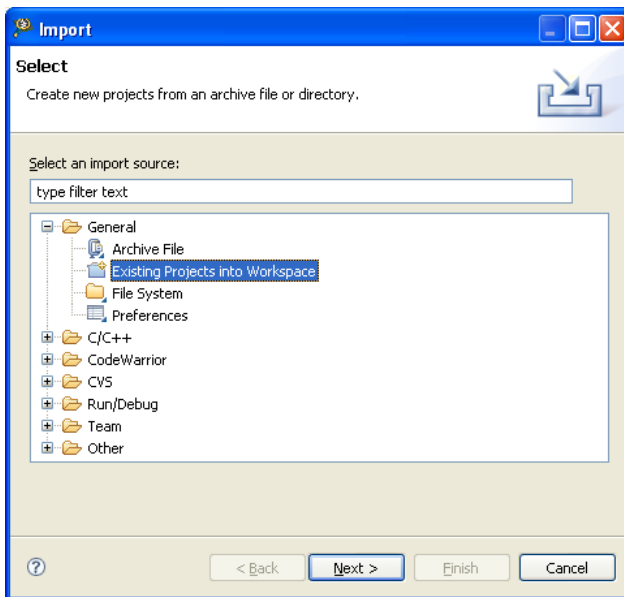


Figure 25. CodeWarrior Import Window

5. Click Next.

The Import Projects window appears as shown in Figure 26. Set the “Select root directory” as the solution root folder (to open a solution with multiple projects) or to that of a single project to open a single project.

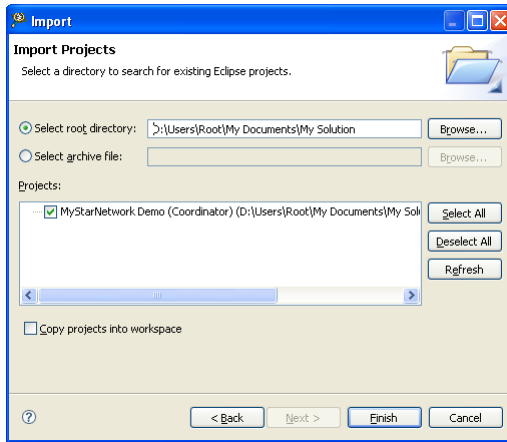


Figure 26. Select Import Root Directory

6. Click the Finish button.

7. If the Remote System Missing dialog message appears as shown in Figure 27, accept the setting by pressing OK.

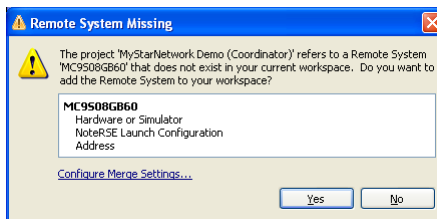


Figure 27. Remote System Missing Window

8. If the Remote Systems Changed window appears as shown in Figure 28, choose the “Replace the current version with the project version.” option.

Users can also configure the workspace to always use the project cached remote system by choosing “Configure Merge Settings” and then choose the “Use the project cached RSE system setting” for the “Select how to handle differing RSE Systems” option.

9. Click OK to accept the settings in the Remote Systems Changed window as shown in Figure 28.

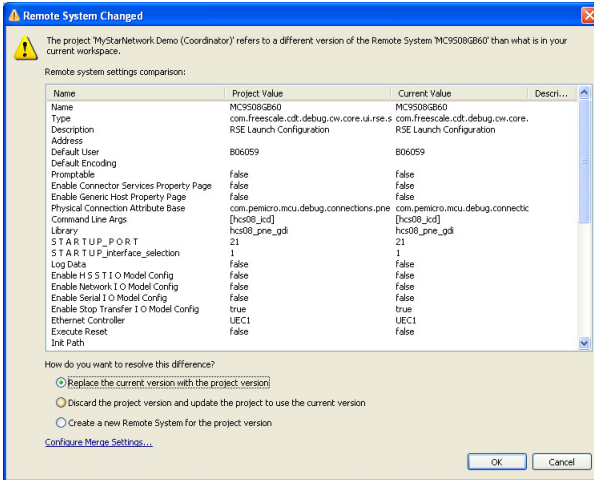


Figure 28. Remote Systems Changed Window

11. The BeeKit solution projects now appears in the CodeWarrior Projects panel as shown in Figure 29. Users can select a project name in the panel and choose Build, Debug, or Run to compile, debug or just load the project to an evaluation board.

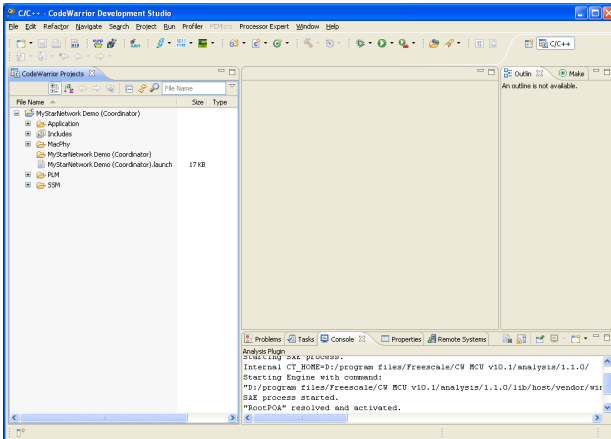


Figure 29. BeeKit Solution Imported into CodeWarrior

12. To import a single project from those exported by BeeKit, bypass Steps 2 through 5 as described in this section by navigating to the project folder and drag-and-drop the `.project` file from the folder to the CodeWarrior window to perform the project import into the current CodeWarrior Workspace.

Importing BeeKit Solutions to IAR Embedded Workbench

To import an exported ARM7 BeeKit solution in IAR Embedded WorkBench perform the following steps:

1. From the IAR menu bar select: File -> Open -> Workspace
2. In the Open Workspace window, navigate to the exported solution folder and choose the exported `*.eww` file.
3. The projects will load to the IAR EWARM GUI as shown in [Figure 30](#).

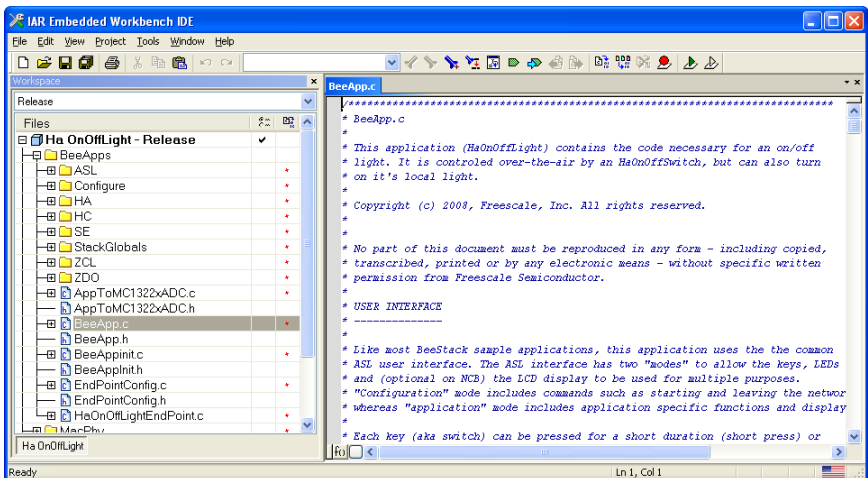


Figure 30. Exported BeeKit Projects in IAR EWARM IDE

Exporting and Loading BeeKit Solutions to IDEs from the BeeKit GUI

BeeKit has the option to automatically open the exported solutions in the respective IDE. To export and open the solution from BeeKit:

1. Depending on the architecture being used, from the menu bar of the BeeKit main window ([Figure 16](#)), click on one of the following:

Solution -> Export and Open Solution in CodeWarrior

Solution -> Export and Open Solution in IAR EWB

Alternatively user can click the IDE icon in the BeeKit tool bar.

The Open Solution window appears as shown in [Figure 31](#).

2. Make sure the correct IDE version is shown in the Choose builder combo box and click OK.

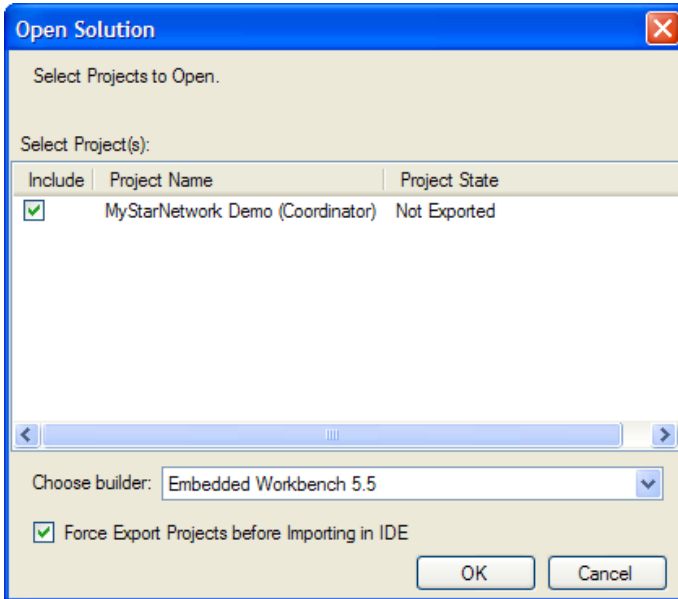


Figure 31. Export Solution Window

3. Ensure that the Project to be exported is checked so that it is included and click the “OK” button. Various messages appear in the message display area of the BeeKit main window as the Solution is exported.
4. The CodeWarrior IDE or Embedded WorkBench IDE is launched and the BeeKit Project is automatically imported.

NOTE

When using the Open Solution option for CodeWarrior 10.1, a new Eclipse workspace is created in the solution folder. The workspace is locked while open in CodeWarrior and another export cannot be performed until the CodeWarrior IDE is closed.



NOTE

For some BeeKit projects, such as the BeeStack and BeeStack Consumer BlackBox project types, CodeWarrior or Embedded Workbench are not launched. For these projects, BeeKit directly generates an embedded firmware image file to the project folder after the export process completes. To use the image file, program it to a development board MCU FLASH using the Freescale Test Tool or CodeWarrior (for all HCS08 based platforms including QE128 and MC1323x) or Test Tool (for ARM7/MC1322x platforms).

Application Development

Users are now ready to import, compile, and load the MAC demonstration code into their hardware. Complete instructions on how to import, compile, and deploy the Project using either the CodeWarrior IDE or the IAR Embedded Workbench for ARM are described in the *BeeKit Wireless Connectivity Toolkit User's Guide* (BKWCTKUG) and the appropriate IDE documentation.

Instructions on how to run the MAC demonstration applications for HCS08 based platforms and ARM7 based platforms are contained in the *802.15.4 Media Access Controller (MAC) MyStarNetworkworkApp User's Guide* (802154MSNAUG).

Instructions on how to run the ZigBee demonstration applications for HCS08 based platforms and ARM7 based platforms that come with BeeStack are contained in the *ZigBee Applications User's Guide* (ZAUGZB2007). To start developing custom BeeStack applications, see the *BeeStack Application Development Guide* (BSADGZB2007).

Instruction on how to run the HCS08 based platform SMAC demonstration applications are contained in the *Running SMAC Demonstration Operations* Application Note (AN3231). For ARM7 based platforms, see the *SMAC Demonstration Application User's Guide* (22xSMACDAUG).

Instruction on how to run the HCS08 based SynkroRF wireless applications for consumer electronics see the *SynkroRF User's Guide* (SYNKROUG).

Instruction on how to run the HCS08 based BeeStack Consumer (ZigBee RF4CE) wireless applications for consumer electronics see the *BeeStack Consumer Application User's Guide* (BSCONAUG).

NOTE

All documents are located on the Freescale web site and in the appropriate sub-folder of the BeeKit Documentation installation folder.



NOTES

Learn More: For more information about
Freescale products, please visit
www.freescale.com/zigbee

Freescale and the Freescale logo are trademarks or registered trademarks of Freescale Semiconductor, Inc. in the U.S. and other countries. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008, 2009.

Doc Number: BKWCTKQSG REV 2.4
Agile Number: 928-22870 (Agile REV 2.3)

