/arrior Development Studio

CodeWarrior Project Importer — Quick Start

The CodeWarrior Project Importer lets you convert a classic CodeWarrior project (`.mcp`) into an Eclipse IDE project.

NOTE In procedures, numbered steps are for advanced users. Lettered steps provide expanded descriptions.

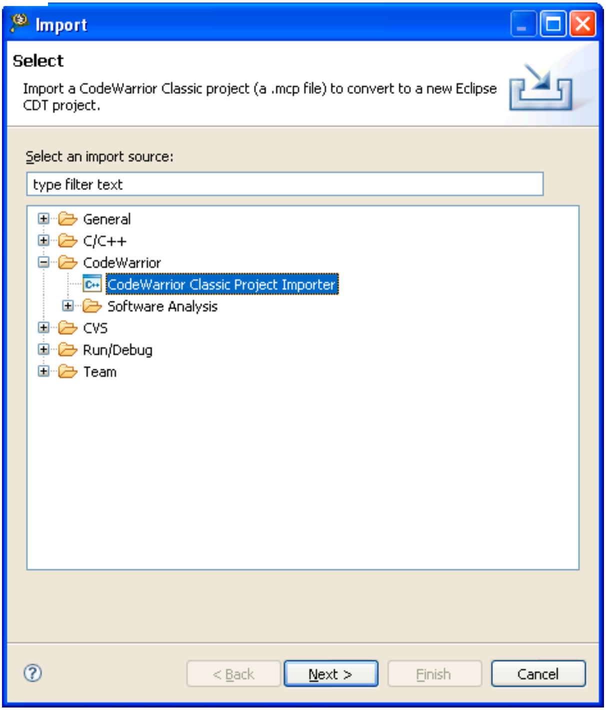
Section A: Importing Classic CodeWarrior Project

To import a classic CodeWarrior project into Eclipse IDE:

1. Select **Start > Programs > Freescale CodeWarrior >** ***your CodeWarrior product*** — The Eclipse-hosted CodeWarrior IDE starts and displays its main window.
2. Select **File > Import** from the CodeWarrior menu bar — The **Import** dialog box appears.

NOTE You can also drag-and-drop a classic CodeWarrior project (`.mcp`) file from Windows explorer to the Eclipse workbench. This approach causes the CodeWarrior Project Importer to start with the correct project name in its project file text box.

Import Dialog Box



Import

Select

Import a CodeWarrior Classic project (a .mcp file) to convert to a new Eclipse CDT project.

Select an import source:

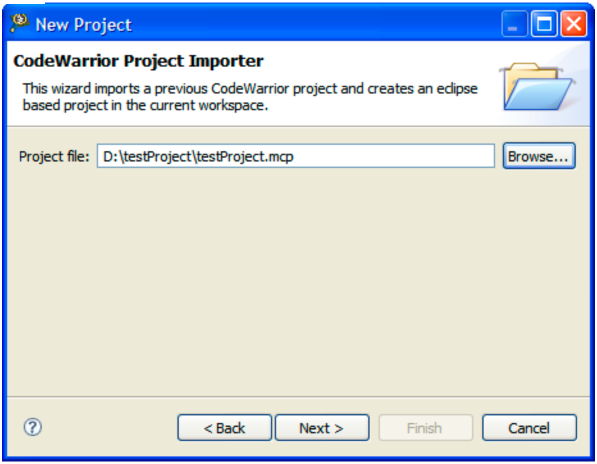
type filter text

- General
- C/C++
- CodeWarrior
  - CodeWarrior Classic Project Importer
  - Software Analysis
- CVS
- Run/Debug
- Team

a. Expand CodeWarrior tree control and select **CodeWarrior Classic Project Importer**.

3. Click **Next** — The first page of the **CodeWarrior Project Importer** wizard appears.

Warrior Project Importer — Select Project File Page



CodeWarrior Project Importer

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

Project file: D:\testProject\testProject.mcp

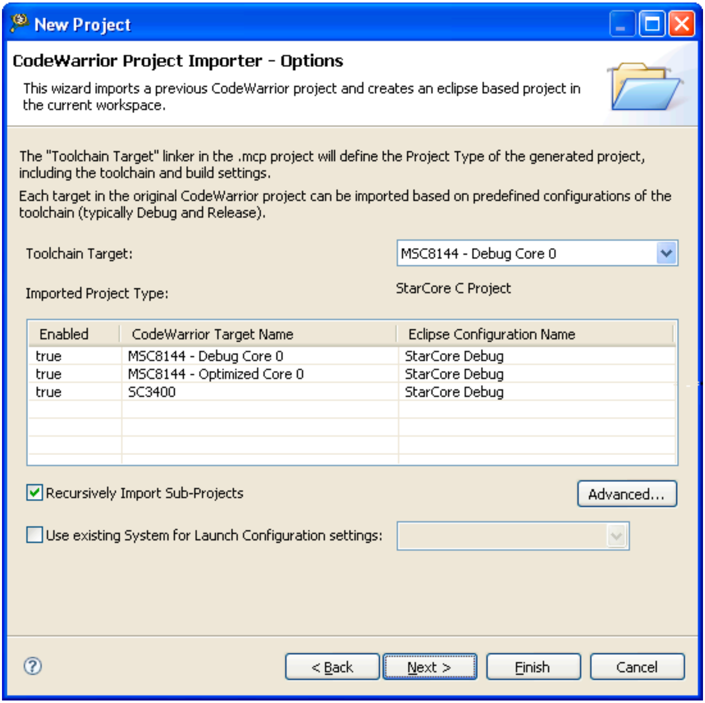
a. Click **Browse** — The **Select the CodeWarrior Project File to Import** dialog box appears.

b. Use this dialog box to navigate to and select the CodeWarrior project (.mcp) file to import.

4. Click **Next** — The **Options** page appears.

NOTE The **Options** page summarizes the build targets information retrieved from the selected project file.

CodeWarrior Project Importer — Options Page



New Project

CodeWarrior Project Importer - Options

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

The "Toolchain Target" linker in the .mcp project will define the Project Type of the generated project, including the toolchain and build settings.

Each target in the original CodeWarrior project can be imported based on predefined configurations of the toolchain (typically Debug and Release).

Toolchain Target: MSC8144 - Debug Core 0

Imported Project Type: StarCore C Project

Enabled	CodeWarrior Target Name	Eclipse Configuration Name
true	MSC8144 - Debug Core 0	StarCore Debug
true	MSC8144 - Optimized Core 0	StarCore Debug
true	SC3400	StarCore Debug

☑ Recursively Import Sub-Projects

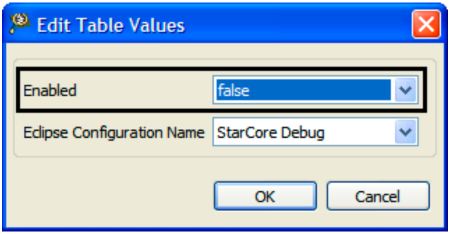
□ Use existing System for Launch Configuration settings:

a. Select a **Toolchain Target** from the listbox — The Build Target Table displays all build targets for the selected toolchain target.

b. To disable the generation of specific configurations, do the following:

i) Click a row in the Build Target table — The **Edit Table Values** dialog box appears.

odeWarrior Project Importer - Edit Table Values



Enabled: false

Eclipse Configuration Name: StarCore Debug

ii) Set Enabled to false.

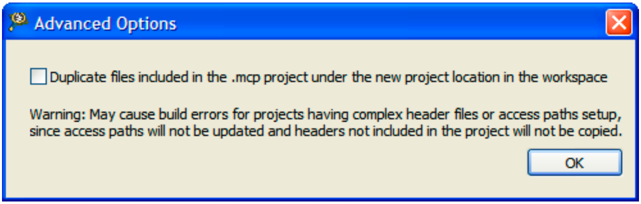
iii) Click **OK** — The **Edit Table Values** dialog box closes.

c. Check the **Recursively Import Sub-Projects** checkbox to import sub-projects as projects along with the main project into the Eclipse-based IDE.

d. If you want to copy all project files under the new Eclipse project directory, click **Advanced** — The **Advanced Options** dialog box appears.

NOTE In normal mode, the CodeWarrior Project Importer does not copy the project files to a newly created Eclipse project. The project files are referenced from their original location.

Advanced Options Dialog Box



□ Duplicate files included in the .mcp project under the new project location in the workspace

Warning: May cause build errors for projects having complex header files or access paths setup, since access paths will not be updated and headers not included in the project will not be copied.

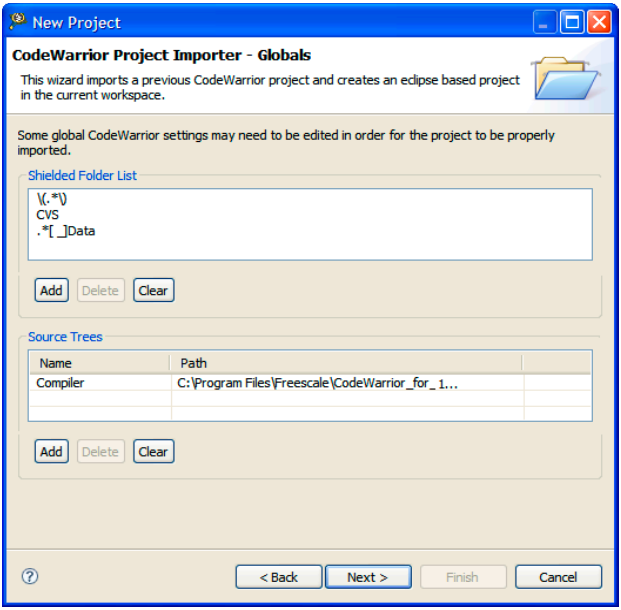
e. If you want to copy all project files under the new eclipse project directory, then check the Duplicate files included in the .mcp project under the new project location in the workspace checkbox.

f. Click **OK** — The **Advanced Options** dialog box closes.

g. Check the **Use existing System for Launch Configuration settings** to select launch configurations from an exisitng system.

xt — The **Globals** page appears.

CodeWarrior Project Importer — Globals Page



CodeWarrior Project Importer - Globals

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

Some global CodeWarrior settings may need to be edited in order for the project to be properly imported.

Shielded Folder List

```
\(.*\)
CVS
.*[ _]Data
```

Source Trees

Name	Path
Compiler	C:\Program Files\Freescale\CodeWarrior_for_1...

a. Modify information in the Shielded Folders List and the Source Trees table using these buttons:

- **Add** — Adds new entry to list.
- **Delete** — Deletes selected item.
- **Clear** — Clears entire list.

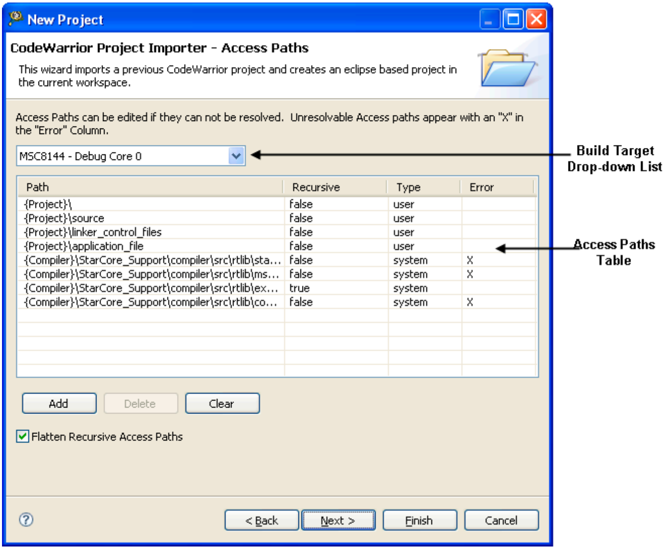
b. Use the **Add**, **Delete**, or **Clear** buttons to modify the Shielded Folder List group — The contents of the folders will be hidden from the IDE's file path resolution operations during a build.

c. Use the **Add**, **Delete**, or **Clear** buttons to modify the Source Trees table — The Source Trees table contains the location of the source trees. If an access path is defined relative to a source tree, the source tree should be listed in this table. The `{Project}` source tree is defined automatically.

5. Click **Next** — The **Access Paths** page appears.

…e **Access Paths** page enables you to add or delete access paths. Access paths are used to search for project files, libraries, runtime support files, and other object files.

CodeWarrior Project Importer — Access Paths Page



CodeWarrior Project Importer - Access Paths

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

Access Paths can be edited if they can not be resolved. Unresolvable Access paths appear with an "X" in the "Error" Column.

MSC8144 - Debug Core 0

Path	Recursive	Type	Error
{Project}\	false	user	
{Project}\source	false	user	
{Project}\linker_control_files	false	user	
{Project}\application_file	false	user	
{Compiler}\StarCore_Support\compiler\src\rtlib\sta…	false	system	X
{Compiler}\StarCore_Support\compiler\src\rtlib\ms…	false	system	X
{Compiler}\StarCore_Support\compiler\src\rtlib\ex…	true	system	
{Compiler}\StarCore_Support\compiler\src\rtlib\co…	false	system	X

☑ Flatten Recursive Access Paths

Build Target Drop-down List

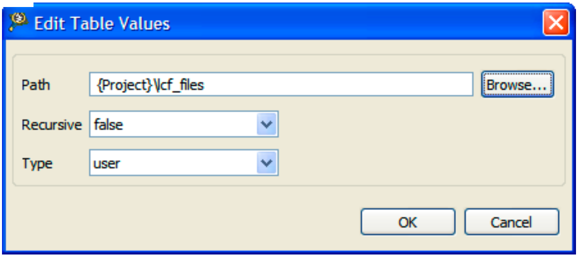
Access Paths Table

a. Select the Build Target from the listbox to modify that target's access paths — The Build Target's access paths appear in the table.

b. Use the **Add**, **Delete**, or **Clear** buttons to edit the Access Paths.

c. Select a row to edit an access path — The **Edit Table Values** dialog box appears.

Edit Table Values Dialog Box



Edit Table Values

Path: {Project}\lcf_files

Recursive: false

Type: user

d. Click **Browse** and navigate to the directory where the access path is searched.

e. Select **true** from the **Recursive** listbox to enable the recursive search of the access path.

f. Select user or system from the Type listbox.

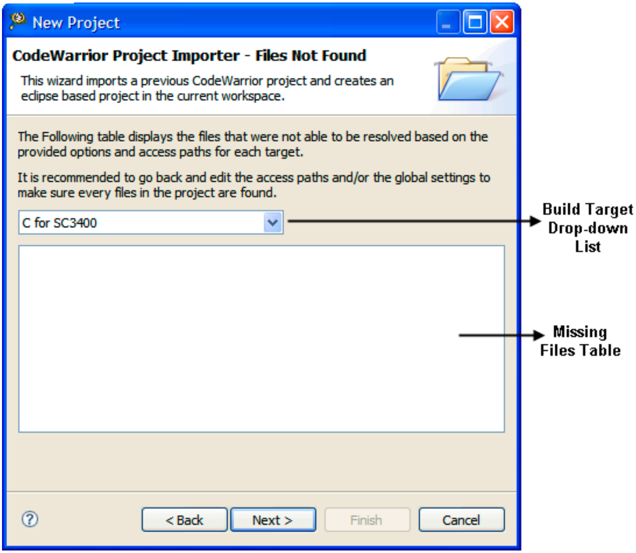
g. Click **OK** — The **Edit Table Values** dialog box closes.

NOTE If the Import Wizard cannot find the specified path, an X appears in the Error column.

h. Check the **Flatten Recursive Access Paths** checkbox to generate a separate Eclipse search path for each subdirectory in a recursive path — Access paths with the Recursive field marked as true will be flattened.

6. Click **Next** — The **Files Not Found** page appears.

eWarrior Project Importer — Files Not Found Page



CodeWarrior Project Importer - Files Not Found

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

The Following table displays the files that were not able to be resolved based on the provided options and access paths for each target.

It is recommended to go back and edit the access paths and/or the global settings to make sure every files in the project are found.

C for SC3400

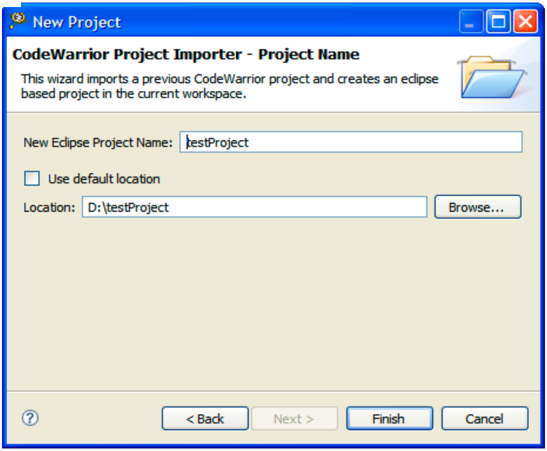
a. Select a Build Target from the listbox — Any files the Build target is missing are listed in the display box.

b. If there are some missing files, click **Back** to adjust the settings in the **Globals** and **Access Paths** pages so that the importer can locate the missing files.

c. Repeat Step until you reduce the number of missing files.

7. Click **Next** — The **Project Name** page appears.

deWarrior Project Importer — Project Name Page



New Project

CodeWarrior Project Importer - Project Name

This wizard imports a previous CodeWarrior project and creates an eclipse based project in the current workspace.

New Eclipse Project Name: testProject

□ Use default location

Location: D:\testProject

a. If you want to specify a new name for the Eclipse-based project, enter the name in the **New Eclipse Project Name** text box.

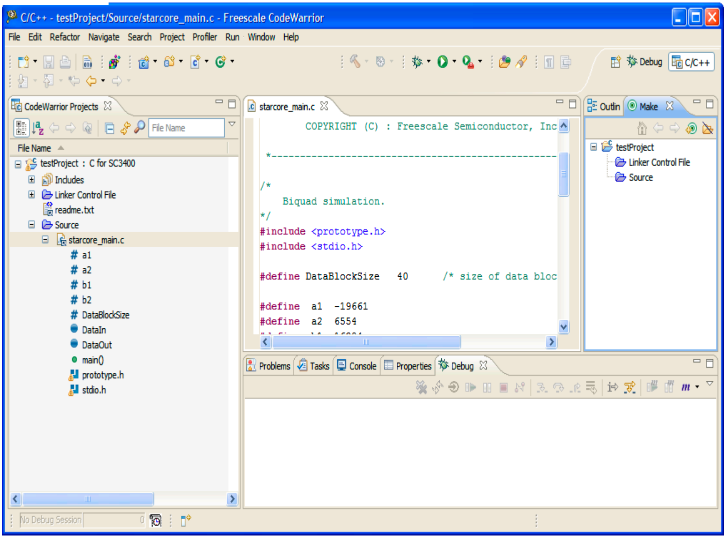
b. If you want the CodeWarrior Project Importer to create the converted project in the Eclipse workspace, check the Use default location checkbox.

NOTE You must specify a folder that does not exist; otherwise, the CodeWarrior Project Importer displays an error message.

8. Click **Finish** — The project importer translates the specified classic CodeWarrior project, and the new Eclipse project appears in the CodeWarrior Project view of the Workbench window.

NOTE Before debugging the new Eclipse project you might need to edit the build and launch configuration settings.

Imported CodeWarrior Project



```
        COPYRIGHT (C) : Freescale Semiconductor, Inc

 *-----------------------------------------------------

/*
    Biquad simulation.
*/
#include <prototype.h>
#include <stdio.h>

#define DataBlockSize   40      /* size of data bloc

#define  a1  -19661
#define  a2  6554
```

Congratulations!

You have successfully converted a classic CodeWarrior project to an Eclipse IDE project.

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.

How to Contact Us

Corporate Headquarters	Freescale Semiconductor, Inc. 6501 William Cannon Drive West Austin, Texas 78735 U.S.A.
World Wide Web	`http://www.freescale.com/codewarrior`
Technical Support	`http://www.freescale.com/support`

Revised: August 16, 2011