

# ARM<sup>®</sup> DS-5<sup>™</sup>

## Using Eclipse

The ARM logo, consisting of the letters "ARM" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right.

# ARM DS-5

## Using Eclipse

Copyright © 2010 ARM. All rights reserved.

### Release Information

The following changes have been made to this book.

#### Change History

Date	Issue	Confidentiality	Change
June 2010	A	Non-Confidential	First release for DS-5

### Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

### Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

### Product Status

The information in this document is final, that is for a developed product.

### Web Address

<http://www.arm.com>

# Contents

## ARM DS-5 Using Eclipse

### Chapter 1      Conventions and feedback

### Chapter 2      Introduction

2.1	About Eclipse .....	2-2
2.2	Integrating ARM plug-ins into a custom Eclipse environment .....	2-3

### Chapter 3      Getting Started with Eclipse

3.1	Launching Eclipse .....	3-3
3.2	Using the welcome screen .....	3-4
3.3	Overview of the workbench window .....	3-6
3.4	Workbench features .....	3-7
3.5	Workspace .....	3-8
3.6	Resources .....	3-10
3.7	Perspectives and views .....	3-14
3.8	Menus .....	3-19
3.9	Toolbars .....	3-20
3.10	Editing source code .....	3-21
3.11	Configuring Eclipse .....	3-23
3.12	Preferences dialog box .....	3-24
3.13	Properties dialog box .....	3-26
3.14	Importing and exporting options .....	3-29
3.15	Using the Import wizard .....	3-30

3.16	Using the Export wizard .....	3-32
3.17	Accessing the dynamic help .....	3-34
3.18	Using ARM-specific cheat sheets .....	3-35
3.19	Installing new features .....	3-37
3.20	Restrictions of use .....	3-38

## Chapter 4

### Working with projects

4.1	Project types .....	4-2
4.2	Creating a new C or C++ project .....	4-4
4.3	Importing an existing Eclipse project .....	4-5
4.4	Accessing the build properties for a specific file or project .....	4-8
4.5	Configuring the C/C++ build settings .....	4-9
4.6	Updating a project to a new toolchain .....	4-12
4.7	Adding a new source file to your project .....	4-13

## Chapter 5

### Working with editors

5.1	Overview of the C/C++ editor .....	5-2
5.2	Overview of the ARM assembler editor .....	5-3

## Chapter 6

### Working with Remote System Explorer

6.1	Overview of the Remote System Explorer .....	6-2
6.2	Adding Remote System Explorer views to the DS-5 Debug perspective ...	6-3
6.3	Remote Systems view .....	6-4
6.4	Remote System Details view .....	6-5
6.5	Remote Scratchpad view .....	6-6
6.6	Terminals view .....	6-7

## Chapter 7

### Terminology, shortcuts and icons

7.1	Terminology .....	7-2
7.2	Keyboard shortcuts .....	7-4
7.3	Menu and toolbar icons .....	7-6

# Chapter 1

## Conventions and feedback

The following describes the typographical conventions and how to give feedback:

### Typographical conventions

The following typographical conventions are used:

- `monospace` Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.
- `monospace` Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.
- monospace italic* Denotes arguments to commands and functions where the argument is to be replaced by a specific value.
- `monospace bold` Denotes language keywords when used outside example code.
- italic* Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
- bold** Highlights interface elements, such as menu names. Also used for emphasis in descriptive lists, where appropriate, and for ARM® processor signal names.

## Feedback on this product

If you have any comments and suggestions about this product, contact your supplier and give:

- your name and company
- the serial number of the product
- details of the release you are using
- details of the platform you are using, such as the hardware platform, operating system type and version
- a small standalone sample of code that reproduces the problem
- a clear explanation of what you expected to happen, and what actually happened
- the commands you used, including any command-line options
- sample output illustrating the problem
- the version string of the tools, including the version number and build numbers.

## Feedback on content

If you have comments on content then send an e-mail to [errata@arm.com](mailto:errata@arm.com). Give:

- the title
- the number, ARM DUI 0480A
- if viewing online, the topic names to which your comments apply
- if viewing a PDF version of a document, the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

ARM periodically provides updates and corrections to its documentation on the ARM Information Center, together with knowledge articles and *Frequently Asked Questions* (FAQs).

## Other information

- *ARM Information Center*,  
<http://infocenter.arm.com/help/index.jsp>
- *ARM Technical Support Knowledge Articles*,  
<http://infocenter.arm.com/help/topic/com.arm.doc.faq>

# Chapter 2

## Introduction

The following topics give an overview of Eclipse, its main features and installation requirements:

### Concepts

- *About Eclipse* on page 2-2
- *Integrating ARM plug-ins into a custom Eclipse environment* on page 2-3.

## 2.1 About Eclipse

Eclipse for DS-5 is an *Integrated Development Environment* (IDE) that combines the Eclipse IDE from the Eclipse Foundation with the compilation and debug technology of the ARM® tools. It also combines the GNU toolchain for ARM Linux targets. You can use it as a project manager to create, build, debug, monitor, and manage projects for ARM targets. It uses a single folder called a workspace to store files and folders related to specific projects.

### 2.1.1 See also

#### Reference

- Chapter 3 *Getting Started with Eclipse*.

#### Other information

- *EclipseFoundation*, <http://eclipse.org>
- *GCC, the GNU Compiler Collection*, <http://gcc.gnu.org>.



## 2.2 Integrating ARM plug-ins into a custom Eclipse environment

The following components are required to use Eclipse for DS-5. If you install Development Studio 5, all the required components are installed for you including the Eclipse IDE.

If you have a custom installation of the Eclipse IDE then you must ensure that you have Eclipse v3.5.

Install the following components in the following order before using Eclipse for DS-5:

1. Download and install J2SE 6.0 or later from Sun Microsystems. You must use a 32-bit version of the JRE even when running on a 64-bit operating system.
2. The *C and C++ Development Tools* (CDT) is a set of plug-ins that integrate the C and C++ build environment into the IDE. You can install these plug-ins as part of the Eclipse+CDT bundle from the Eclipse website. CDT v6.0 is required if you want to use the latest ARM® plug-ins.
3. Install Development Studio 5 in a separate location from your custom Eclipse installation. Search for the folder *install\_directory\sw\eclipse* and take a copy of the dropins folder then place it under the eclipse folder in your custom installation.
4. If you want the custom installation of Eclipse to integrate with the GNU toolchain, you must install the latest GNU tools specific for ARM.

---

### Note

Eclipse for DS-5 incorporates the GNU toolchain for ARM Linux targets that corresponds to GCC version 4.4.1.

---

### 2.2.1 See also

#### Tasks

- *Launching the debugger from Eclipse* on page 2-6.

#### Other information

- *Java from Sun Microsystems*, <http://www.java.com>
- *Eclipse*, <http://www.eclipse.org>
- *Sourcery G++ for ARM GNU/Linux*, <http://www.codesourcery.com>.



# Chapter 3

## Getting Started with Eclipse

The following topics describe how to get started with Eclipse, the C/C++ perspective, the DS-5 Debug perspective, and associated features:

### Tasks

- *Launching Eclipse* on page 3-3
- *Using the welcome screen* on page 3-4
- *Overview of the workbench window* on page 3-6
- *Using the Import wizard* on page 3-30
- *Using the Export wizard* on page 3-32
- *Accessing the dynamic help* on page 3-34
- *Using ARM-specific cheat sheets* on page 3-35
- *Installing new features* on page 3-37.

### Reference

- *Workbench features* on page 3-7
- *Workspace* on page 3-8
- *Resources* on page 3-10
- *Perspectives and views* on page 3-14
- *Menus* on page 3-19

- *Toolbars* on page 3-20
- *Editing source code* on page 3-21
- *Configuring Eclipse* on page 3-23
- *Preferences dialog box* on page 3-24
- *Properties dialog box* on page 3-26
- *Importing and exporting options* on page 3-29
- *Restrictions of use* on page 3-38.

## 3.1 Launching Eclipse

To launch Eclipse:

- On Windows, select **Start** → **All Programs** → **ARM DS-5** → **Eclipse for DS-5**.
- On Linux:
  1. Add *install\_directory/bin* to your PATH environment variable. If it is already configured then you can skip this step.
  2. Open Unix bash shell.
  3. Enter *eclipse* at the prompt.

### 3.1.1 See also

#### Concepts

- *Using the welcome screen* on page 3-4
- *Overview of the workbench window* on page 3-6.

#### Reference

- *Workbench features* on page 3-7
- *Workspace* on page 3-8
- *Importing and exporting options* on page 3-29.

## 3.2 Using the welcome screen

When Eclipse launches for the first time, the Workspace Launcher dialog box opens enabling you to select your workspace, accept the default settings and click **OK**. You are then presented with the welcome screen containing an overview of the main development environment and links to other pages to help you get started.

———— **Note** ————

Some of the links require an active internet connection.



Figure 3-1 Welcome screen

To access the main development environment click on the close icon in the **DS-5 Home** tab or click on the **Go to the Workbench** link on the welcome page. You can return to the welcome screen at any time by selecting **DS-5 Home** from the **Help** menu.

### 3.2.1 See also

#### Concepts

- *Overview of the workbench window* on page 3-6.

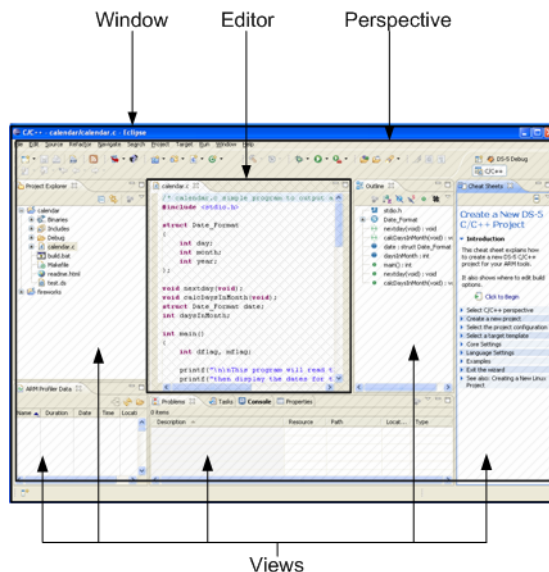
#### Reference

- *Workbench features* on page 3-7
- *Workspace* on page 3-8
- *Importing and exporting options* on page 3-29.

### 3.3 Overview of the workbench window

A typical workbench window contains a collection of views that are associated with a specific perspective. For example the Project Explorer view, C/C++ editor, Progress view, Console view, and Problems view are all associated with the C/C++ perspective. Each perspective automatically adds the most useful views when you select it.

Development Studio 5 uses the C/C++ and DS-5 Debug perspectives.



**Figure 3-2 Typical workbench window**

To close the workbench window and exit, select **Exit** from the **File** menu or click on the close icon in the top corner of the window. On exit, Eclipse saves your settings automatically so that when you next open it, the window returns to the same perspectives and views.

#### 3.3.1 See also

##### Tasks

- *Using the welcome screen* on page 3-4.

##### Reference

- *Workbench features* on page 3-7
- *Workspace* on page 3-8
- *Importing and exporting options* on page 3-29.



## 3.4 Workbench features

The workbench is the main development environment where you can manage individual projects, associated sub-folders, and source files. Each workbench window is linked to one workspace. If you want to use different workspaces at the same time, you can launch several workbench windows and link each one to a different workspace.

This section describes the main workbench features:

**Editors** Editors enable you to view and modify the content of a file, for example source files. The tabs in the editor area show files that are currently open for editing.

### Menus and Toolbars

The main menu and toolbar are located at the top of the Eclipse window. Other toolbars associated with specific features are located at the top of each perspective or view.

**Perspectives** Perspectives define the layout of your selected views and editors in Eclipse. They also have their own associated menus and toolbars.

**Resources** Resources are projects, files, and folders that exist in your workbench.

**Views** Views provide related information corresponding to the active file in the editor. They also have their own associated menus and toolbars.

**Workspace** Workspace is an area designated on your file system to store files and folders related to your workbench projects and also your personal workbench settings.

### 3.4.1 See also

#### Tasks

- *Using the welcome screen* on page 3-4
- Chapter 5 *Working with editors*.

#### Concepts

- *Overview of the workbench window* on page 3-6.

#### Reference

- *Workspace* on page 3-8
- *Resources* on page 3-10
- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Toolbars* on page 3-20.

## 3.5 Workspace

The workspace is an area designated on your file system to store files and folders related to your workbench projects and also your personal workbench settings.

---

### Note

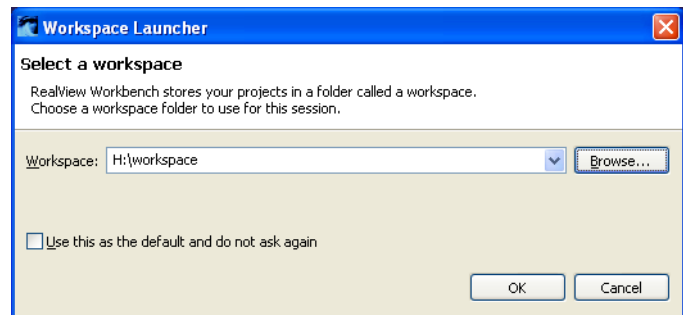
It is recommended that you select a dedicated workspace folder for your workbench projects only. If you select an existing folder containing resources that are not related to workbench projects, you cannot access them in Eclipse. These resources might also cause a conflict later when you create and build projects.

Changes to the customized settings in the **Window** → **Preferences** dialog box are saved in your workspace. If you select a different workspace then these settings might be different.

---

When Eclipse launches for the first time, the Workspace Launcher dialog box opens enabling you to select your workspace.

On subsequent launches the last saved workspace is shown as the default selection in the drop-down menu. You can select another workspace by clicking on the down arrow or the **Browse...** button.



**Figure 3-3 Workspace launcher dialog box**

If required, select the checkbox to disable the Workspace Launcher dialog box from opening on subsequent launches.

---

### Note

You can change the default workspace at any time by selecting **Switch Workspace...** from the **File** menu.

---

Alternatively, to open Eclipse and automatically link to a specific workspace you can use the `-data` command-line argument. For example:

```
eclipse -data h:\workspace
```

### 3.5.1 Editing files outside Eclipse

Project sub-folders and files can be edited even when Eclipse is not running. When you next launch Eclipse, the default preferences enable the relevant views to refresh and update. Alternatively if you change the default preferences to disable auto-refresh, you can click on the updated sub-folder or file in the Project Explorer view and select **Refresh** from the **File** menu.

### 3.5.2 See also

#### Tasks

- *Using the welcome screen* on page 3-4.
- Chapter 5 *Working with editors*.

#### Reference

- *Overview of the workbench window* on page 3-6
- *Workbench features* on page 3-7
- *Resources* on page 3-10
- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Toolbars* on page 3-20.

## 3.6 Resources

A resource is a generic term used to describe a project, file, folder or a combination of these. Resources exist in Eclipse but might not always exist in the workspace. There are three types of resources:

- |                |   |
|----------------|---|
| <b>Project</b> | A project is displayed in the Project Explorer view and can be stored within the workspace folder or can be a linked resource.<br><br>A project must exist in Eclipse before other resources can be imported or linked to that project. The project creation process creates additional configuration files and folders, for example, build properties. These additional files and folders must not be edited or deleted. |
| <b>Folder</b>  | A folder is displayed in the Project Explorer view and can be located within the project folder or can be a linked resource.  |
| <b>File</b>    | A file is displayed in the Project Explorer view and can be located within the project folder or can be a linked resource.  |

### 3.6.1 Linked resources

Resources can be shared between projects or they can exist in the file system outside of your selected workspace. To do this a link must be created within Eclipse.

———— **Note** ————

A linked file or folder must have a project as its parent resource.

Deleting, moving or copying a linked resource only affects the link in your workspace and not the resource that it links to. However, deleting a child resource from within a linked folder also deletes it from the file system!

#### Linked file

To link an existing file to a project in your workspace instead of copying it, you can use the advanced options of the New File wizard. By default the advanced options are not visible, click on the <<**Advanced** button to reveal them. A path variable can also be used to reference a file. For more information on referencing with path variables, use the dynamic help.

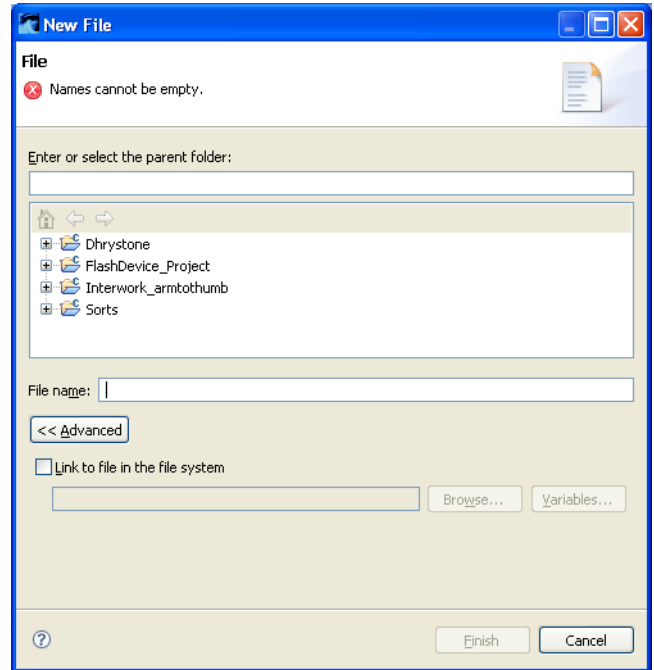
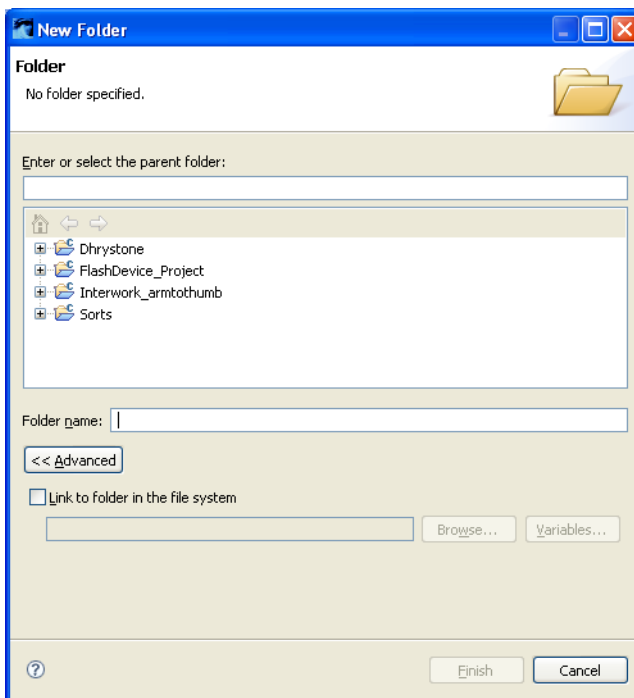


Figure 3-4 Linked file

### Linked folder

To link an existing folder to a project in your workspace instead of copying it, you can use the advanced options of the New Folder wizard. By default the advanced options are not visible, click on the **<<Advanced** button to reveal them. A path variable can also be used to reference a file. For more information on referencing with path variables, use the dynamic help.

**Figure 3-5 Linked folder**

### Linked project

Eclipse uses the Import wizard to create a link to an existing project. This can be useful if you have a central folder with shared projects. By default, the **Copy projects into workspace** checkbox is selected to ensure that a copy of your project is placed in your current workspace. If you want to leave your project where it is and create a link to it, you must deselect this option.

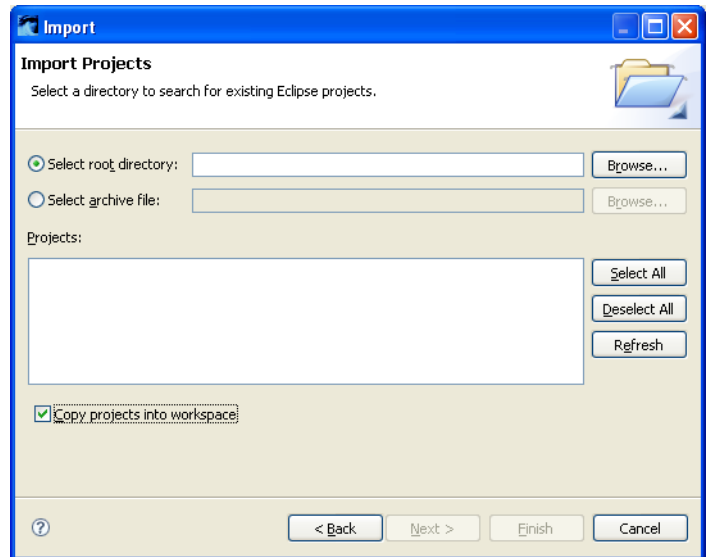


Figure 3-6 Linked project

### Disabling the use of linked resources

You can disable the use of linked resources by changing the **General** → **Workspace** → **Linked Resources** settings in the Preferences dialog box.

### 3.6.2 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Preferences dialog box* on page 3-24.

### 3.7 Perspectives and views

The main workbench window contains one or more perspectives and each perspective contains one or more views.

#### 3.7.1 Perspectives

Perspectives can be opened and customized using the options from the **Window** menu or you can use the perspective toolbar. By default a perspective opens in the same window, however, you can change the default settings in the Preferences dialog box.

Eclipse for DS-5 focuses on the C/C++ perspective and the DS-5 Debug perspective. You can change perspective by using the perspective toolbar or you can select **Window** → **Open perspective** from the main menu.

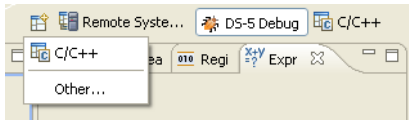


Figure 3-7 Changing perspective using the toolbar menu

#### 3.7.2 Views

A view is a small visual component within Eclipse to navigate through resources or to display specific properties. An editor is a special type of view that enables viewing and editing of source files.

Views can be moved or docked as applicable by dragging and dropping them into position. Double-clicking on a view tab toggles the maximize/restore options for that view or you can use the relevant toolbar icons. To reset all the views in a perspective to the initial settings and layout you can click on **Reset Perspective...** in the **Window** menu.

The following views are useful when building and debugging C/C++ applications:

**Editors** Editors are associated with specific file types and open the related editor view automatically when you open an editable file from the Project Explorer view.

**General views**

<b>Console</b>	Displays output messages when building projects and also execution messages when running or debugging an application.
----------------	---



<b>Outline</b>	Displays a structured list of C/C++ elements in the active file. Clicking on an element changes the editor focus to the position of that element in the active file.
<b>Problems</b>	Displays error messages encountered during a build. Selecting an error message opens the associated file and moves the focus to the line causing the problem.
<b>Progress</b>	Displays the progress meter when a task is in progress.
<b>Project Explorer</b>	Provides a hierarchical view of resources relevant to C/C++ project files. Right-clicking on a resource produces a context menu for specific tasks.
<b>Properties</b>	Displays names and values for the selected item. For example, the last modified time/date for a file.
<b>Help views</b>	
<b>Help</b>	Displays dynamic help for the selected feature.
<b>Cheat Sheets</b>	Displays step by step instructions to help you complete a specific task. Use the view menu to access the list of cheat sheets.
<b>DS-5 Debugger views</b>	
<b>Breakpoints</b>	Displays all the breakpoints and the respective locations in source code or target memory. Using this view you can add, edit, delete, enable or disable breakpoints.
<b>Commands</b>	Provides access to the scripting functionality of the debugger. Most debugger actions are echoed here and you can also execute DS-5 Debugger commands.
<b>Debug Control</b>	Contains a tree hierarchy showing debug connections, threads and stack frames.
<b>Disassembly</b>	Displays disassembled code.
<b>Expressions</b>	Enables you to view and edit expressions.
<b>History</b>	Provides a historical log of all the debugger commands executed during this debug session.
<b>Memory</b>	Enables you to view and edit the contents of memory.

<b>Registers</b>	Enables you to view and edit registers associated with the selected stack frame. When your application stops, modified values are highlighted.
<b>Screen</b>	Enables you to display a block of memory as a graphical image. For example, a frame buffer.
<b>Scripts</b>	Provides access to run and edit script files.
<b>Semihosting Console</b>	
	Enables you to interact with the I/O capabilities provided by the semihosting implementation in the ARM® C libraries.
<b>Target</b>	Displays a list of capabilities for the target device that is currently connected to the debugger.
<b>Variables</b>	Enables you to view and edit variables associated with the selected stack frame.

### Remote Systems views

#### Remote Scratchpad

Enables you to copy and paste or drag and drop local resources to an electronic clipboard for use at a later point in time.

#### Remote Systems

Enables you to create connections and access resources on the host workstation and remote targets.

Other views can be added to a perspective but they might not be compatible with all perspectives. To add a view to a perspective:

1. Ensure that you are in the relevant perspective.
2. Select **Window** → **Show View** → **Other...** to open the Show View dialog box.
3. Select the required view.
4. Click **OK**.

For more information on the other views not listed here, use the dynamic help.

### 3.7.3 Trim bar

Minimized views are placed in the trim bar at the edge of the current perspective. By default the trim bar is on the right-hand margin of the perspective but you can drag and drop it to another location if you prefer. In the trim bar, you can double click on a view to restore it or click on the restore icon to restore all the minimized views.

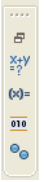


Figure 3-8 Minimized views in the trim area

### 3.7.4 Fast view bar

Views can be docked in a Fast View bar at the edge of the current perspective. You can move the Fast view bar to any edge by clicking on the vertical separator line and dragging it to a new location. You can add a view to the Fast View bar by clicking on the **Show view as a fast view** icon and selecting a view from the resultant list.



Figure 3-9 Fast view bar

### 3.7.5 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- Chapter 5 *Working with editors*.

#### Reference

- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Preferences dialog box* on page 3-24
- *Breakpoints view* on page 10-3
- *Commands view* on page 10-16
- *Debug Control view* on page 10-19
- *Disassembly view* on page 10-24
- *History view* on page 10-31
- *Memory view* on page 10-33
- *Registers view* on page 10-36
- *Screen view* on page 10-39
- *Scripts view* on page 10-41
- *Semihosting view* on page 10-43

- *Variables view* on page 10-48
- *Perspective icons* on page 7-7
- *View markers* on page 7-7.

## 3.8 Menus

The main menu is located at the top of the Eclipse window and can be customized to your personal preferences. The contents might vary depending on the installed plug-ins and also the active perspective.

Eclipse supports the following options from the main menu:

<b>File</b>	Enables you to create, save, close, print, import, and export resources. You can also manage project and file property settings.
<b>Edit</b>	Enables you to cut, copy, paste, find, and replace text within a resource.
<b>Navigate</b>	Enables you to navigate and quickly find specific resources.
<b>Search</b>	Provides an advanced filter for searching through resources.
<b>Project</b>	Enables you to manage project build configurations and perform specific builds. You can also customize the build settings for ARM® tools.
<b>Run</b>	Enables you to run, send to, debug or configure external tools. You can also manage breakpoints and watchpoints.
<b>Window</b>	Enables you to open, close, and customize perspectives, views and editors.
<b>Help</b>	Provides documentation on Eclipse and the ARM tools. You can also access the ARM cheat sheets and software updates.

Right-clicking on a resource produces a context menu for specific tasks. For more information on the other menu options not listed here, use the dynamic help.

### 3.8.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Perspectives and views* on page 3-14
- *Menus*
- *Toolbars* on page 3-20
- *Preferences dialog box* on page 3-24
- *Menu and toolbar icons* on page 7-6.

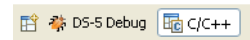
## 3.9 Toolbars

The main toolbar is located at the top of the Eclipse window and can be customized to your personal preferences. The following figure shows the main workbench toolbar for the C/C++ perspective. The contents vary depending on the installed plug-ins and also the active perspective.

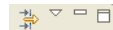


**Figure 3-10 Workbench toolbar**

Other toolbars associated with specific features are located at the top of each perspective or view.



**Figure 3-11 Perspective toolbar**



**Figure 3-12 View toolbar**

### 3.9.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Preferences dialog box* on page 3-24
- *Menu and toolbar icons* on page 7-6.

## 3.10 Editing source code

You can use the editors provided with Eclipse to edit your source code or you can use an external editor. If you work with an external editor you must refresh Eclipse to synchronize the views with the latest updates. To do this, in the Project Explorer view, you can click on the updated project, sub-folder, or file and select **Refresh** from the **File** menu. Alternatively you can enable the automatic refresh option by selecting **General** → **Workspace** → **Refresh automatically** in the Preferences dialog box. See the *Preferences dialog box* on page 3-24 for more information.

When you open a file in Eclipse, a new editor tab appears with the name of the file. An edited file displays an asterisk (\*) in the tab name to show that it has unsaved changes.

When you have two or more editor tabs open, you can tile them for side-by-side viewing by clicking on a tab and dragging it over an editor border.

In the left-hand margin of each editor tab you can find a vertical bar that displays markers relating to the active file. See *View markers* on page 7-7 for more information.

### 3.10.1 Navigating

There are several ways to navigate to a specific resource within Eclipse. You can use the Project Explorer view to open a resource by browsing through the resource tree and double-clicking on a file. An alternative is to use the keyboard shortcuts or use the options from the **Navigate** menu.

The **Navigate** menu enables you to locate a resource by pattern matching using the Go To option or you can use the Open Resource option to directly open a file in an editor.

### 3.10.2 Searching

To locate information or specific code contained within one or more files in Eclipse, you can use the options from the **Search** menu. Textual searching with pattern matching and filters to refine the search fields are provided in a customizable Search dialog box. You can also open this dialog box from the main workbench toolbar.

### 3.10.3 Content assist

The C/C++ editor, ARM® assembler editor, and the DS-5 Debugger Commands view provide content assistance at the cursor position to auto-complete the selected item. Using the Ctrl+Space keyboard shortcut produces a small dialog box with a list of valid options to choose from. You can shorten the list by partially typing a few characters before using the keyboard shortcut. From the list you can use the Arrow Keys to select the required item and then press the Enter key to insert it.

### 3.10.4 Bookmarks

Bookmarks can be used to mark a specific position in a file or mark an entire file so that you can return to it quickly. To create a bookmark, select a file or line of code that you want to mark and select **Add Bookmark** from the **Edit** menu. The Bookmarks view displays all the user defined bookmarks and can be accessed by selecting **Window** → **Show View** → **Bookmarks** from the main menu. If the Bookmarks view is not listed then select **Others...** for an extended list.

To delete a bookmark, open the Bookmarks view, click on the bookmark that you want to delete and select **Delete** from the **Edit** menu.

### 3.10.5 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Perspectives and views* on page 3-14
- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.



## 3.11 Configuring Eclipse

Eclipse can be customized to your own settings by changing the layout, key bindings, file associations, and color schemes.

Projects and files can also be configured to use the build system in different ways by modifying the properties for the selected resource.

Perspectives can be opened and customized using the options from the **Window** menu or you can use the perspective toolbar. By default a perspective opens in the same window, however, you can change the default settings in the Preferences dialog box.

Views can be moved or docked as applicable by dragging and dropping them into position. Double-clicking on the title bar of a view toggles the maximize/restore options or you can use the relevant toolbar icons. If you have minimized a view you can restore it by clicking on the restore icon in the toolbar at the side of the perspective. By default this is on the right-hand side of the perspective but you can drag and drop it to another location if you prefer. To reset all the views in a perspective to the initial settings and layout you can click on **Reset Perspective...** in the **Window** menu.

### 3.11.1 See also

#### Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Properties dialog box* on page 3-26
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 3.12 Preferences dialog box

Workbench settings can be customized using the Preferences dialog box. You can access this dialog box by selecting **Preferences...** from the **Window** menu. Changes to these settings are saved in the current workspace. If you want to copy your workbench settings to another workspace, use the Export wizard.

The contents of the preferences hierarchy tree include the following:

<b>General</b>	Controls the workspace, perspectives, editors, build order, linked resources, file associations, path variables, background operations, keyboard and mouse settings.
<b>C/C++</b>	Controls the C/C++ environment settings, CDT build variables, syntax formatting, and default project wizard settings.
<b>Default Device Properties</b>	
	Controls the default device properties.
<b>DS-5 Debugger</b>	Controls the default debugger settings and also the presentation and formatting for the ARM® assembler.
<b>Help</b>	Controls how the context help is displayed.
<b>Install/Update</b>	Controls the update history, scheduler, and policy.
<b>Remote Systems</b>	Controls the settings used by the Remote System Explorer.
<b>Run/Debug</b>	Controls the default perspectives, breakpoint, build, and launch settings before running and debugging.

For more information on the other options not listed here, use the dynamic help.

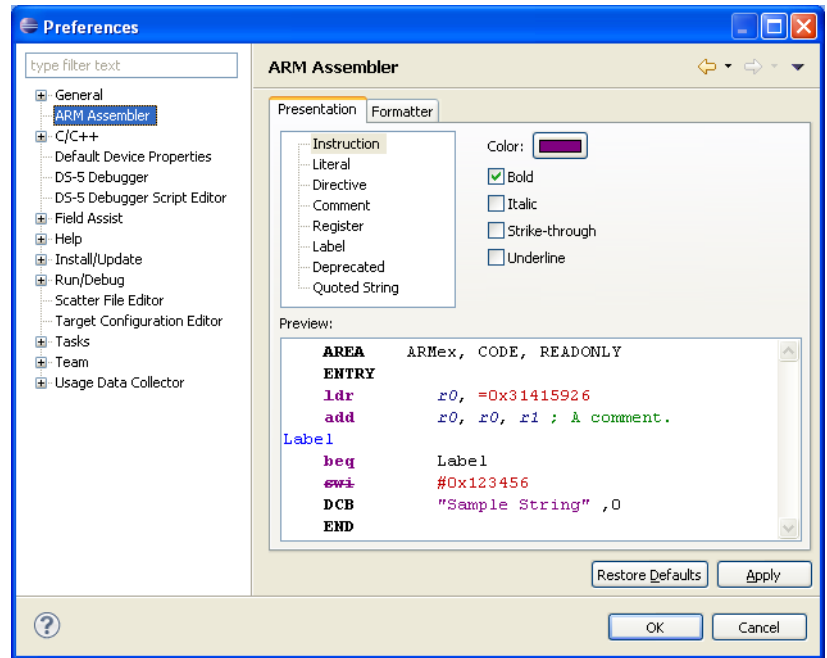


Figure 3-13 Window preferences dialog box

### 3.12.1 See also

#### Reference

- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Menus* on page 3-19
- *Perspectives and views* on page 3-14
- *Toolbars* on page 3-20
- *Configuring Eclipse* on page 3-23
- *Properties dialog box* on page 3-26
- *Importing and exporting options* on page 3-29
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 3.13 Properties dialog box

Project settings can be customized using the Properties dialog box. You can access this dialog box by selecting a project and then selecting **Properties...** from the **Project** menu. Changes to the customized settings are saved in the project folder in your workspace. You can also customize the C/C++ properties for a single file for example, if you want to apply a specific compiler option to a file during the build.

---

**Note**

---

If you specify different options for a single file, it overrides the options specified in the project configuration panels that apply to all related source files.

---

The contents of the properties hierarchy tree for a project include the following:

<b>Resource</b>	Displays the resource location, modification state, and file type.
<b>Builders</b>	Controls builders available for the selected project.
<b>C/C++ Build</b>	Controls the environment, build, and tool chain settings for the active configuration.
<b>C/C++ General</b>	Controls documentation, file types, indexer and path/symbol settings.
<b>Project References</b>	Controls project dependencies.

For more information on the other options not listed here, use the dynamic help.

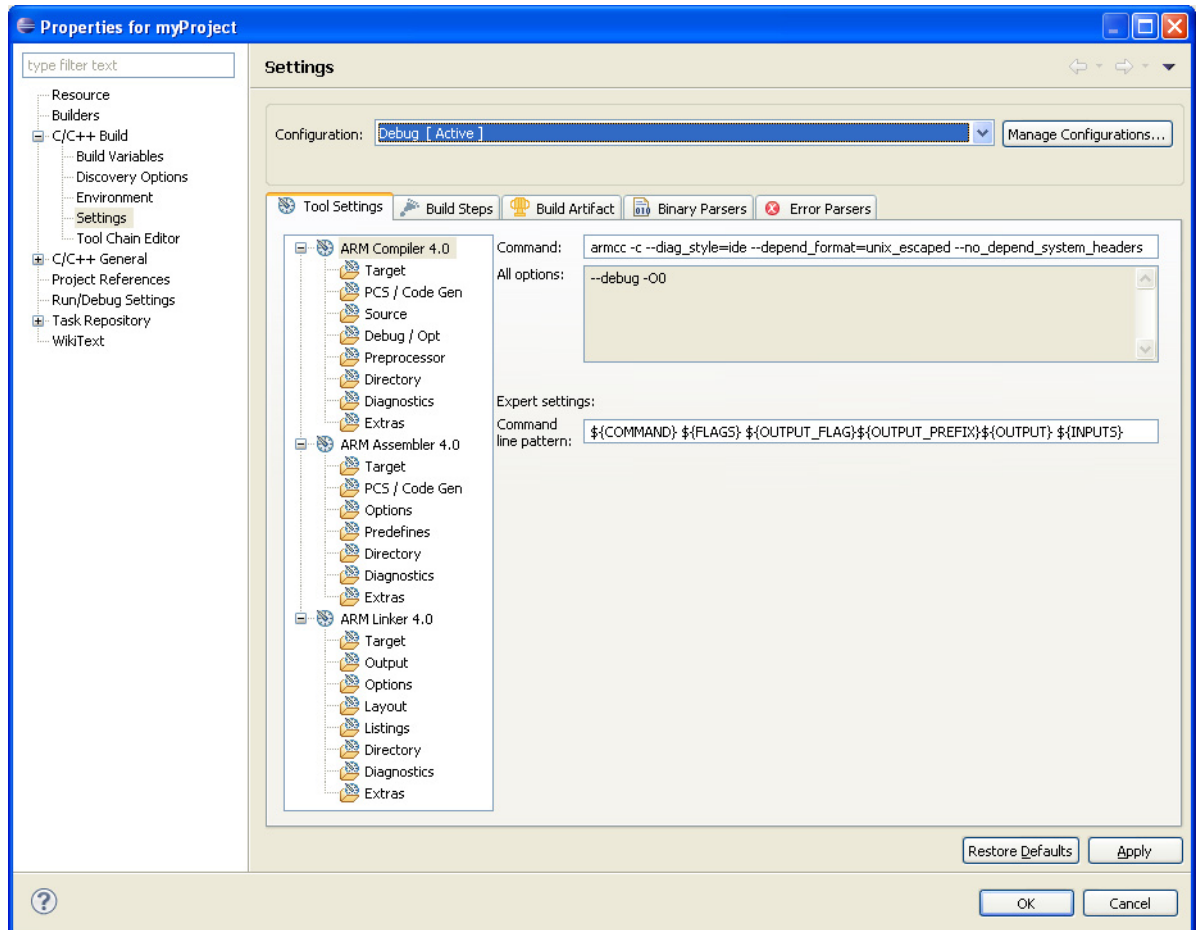


Figure 3-14 Project properties dialog box

### 3.13.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34

#### Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14

- *Configuring Eclipse* on page 3-23
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 3.14 Importing and exporting options

A resource must exist in a project within Eclipse before you can use it in a build. If you want to use an existing resource from your file system in one of your projects, the recommended method is to use the Import wizard. To do this, select **Import...** from the **File** menu.

If you want to use a resource externally from Eclipse, the recommended method is to use the Export wizard. To do this, select **Export...** from the **File** menu.

There are several options available in the import and export wizards:

**General** This option enables you to import and export the following:

- archive zip files
- selected source files and project sub-folders
- files containing workbench preference settings.

You can also import existing projects into your workspace.

**C/C++** This option enables you to import the following:

- C/C++ executable files
- C/C++ project settings.

You can also export C/C++ project settings.

### Remote Systems

This option enables you to transfer files between the local host and the remote target.

For information on the other options not listed here, use the dynamic help.

### 3.14.1 See also

#### Tasks

- *Using the Import wizard* on page 3-30
- *Using the Export wizard* on page 3-32
- *Accessing the dynamic help* on page 3-34.

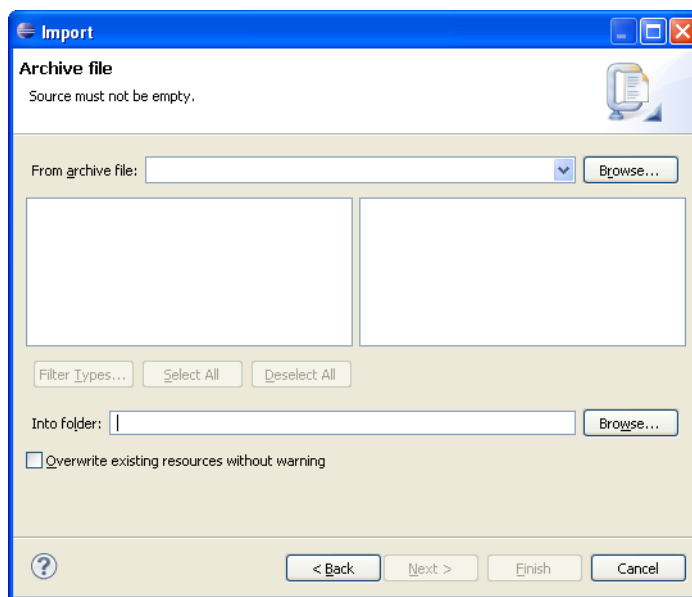
## 3.15 Using the Import wizard

The Import wizard can be used to import complete projects, source files and, project sub-folders in addition to breakpoint and preference settings. Select **Import...** from the **File** menu.

This section focuses on projects, source files and project sub-folders.

To import a complete project either from an archive zip file or an external folder from your file system, you must use the Existing Projects into Workspace or CodeWarrior Project exported as XML wizard. This ensures that the relevant workbench files are also imported into your workspace.

Individual source files and project sub-folders can be imported using either the Archive File or File System wizard. Both options produce a dialog box similar to the example shown below. Using the options provided you can select the required resources and specify the relevant options, filename, and destination path.



**Figure 3-15 Typical example of the import wizard**

With the exception of the Existing Projects into Workspace wizard, files and folders are copied into your workspace when you use the Import wizard. To create a link to an external file or project sub-folder you must use the New File or New Folder wizard.



### 3.15.1 See also

#### Tasks

- *Using the Export wizard* on page 3-32
- *Accessing the dynamic help* on page 3-34.

#### Reference

- *Linked resources* on page 3-10
- *Importing and exporting options* on page 3-29.

## 3.16 Using the Export wizard

The Export wizard can be used to export complete projects, source files and, project sub-folders in addition to breakpoint and preference settings. Select **Export...** from the **File** menu.

This section focuses on projects, source files and project sub-folders.

Exporting a complete project, source file or project sub-folder uses the same process. If you want to create a zip file you can use the Archive File wizard, or alternatively you can use the File System wizard. Both options produce a dialog box similar to the example shown below. Using the options provided you can select the required resources and specify the relevant options, filename, and destination path.

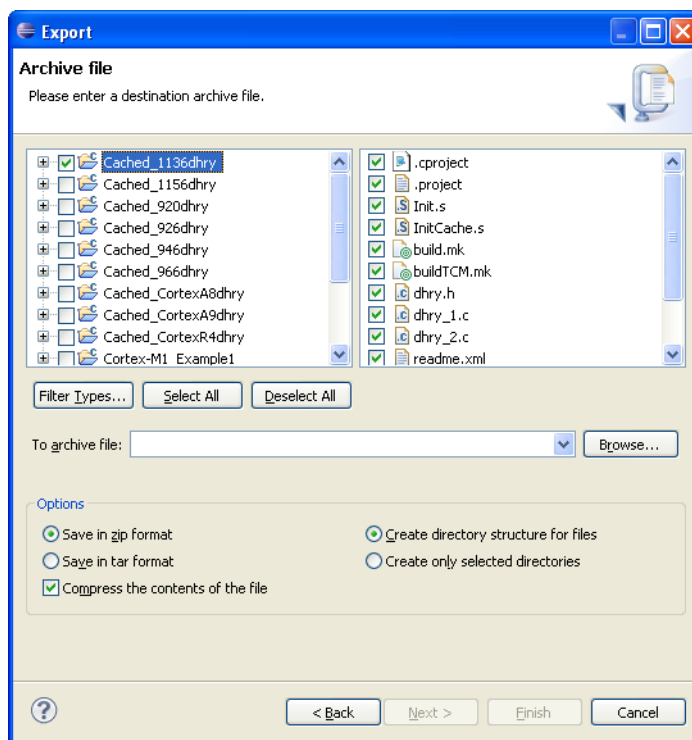


Figure 3-16 Typical example of the export wizard

### 3.16.1 See also

#### Tasks

- *Using the Import wizard* on page 3-30

- *Accessing the dynamic help* on page 3-34.

### **Reference**

- *Linked resources* on page 3-10
- *Importing and exporting options* on page 3-29.

## 3.17 Accessing the dynamic help

To access the dynamic help for a specific workbench feature you must:

1. Click on an editable field for the feature that you want to use.
2. Do one of the following:
  - Click on the question mark icon
  - Select **Dynamic Help** from the **Help** menu.
  - Use a keyboard shortcut:
    - On Windows, **F1** key
    - On Linux , **Shift+F1** key combination.

Dynamic help for the selected feature appears in the About panel at the top of the Related Topics view.

---

**Note**

Other possible search results are listed in the Dynamic Help panel at the bottom of the Related Topics view.

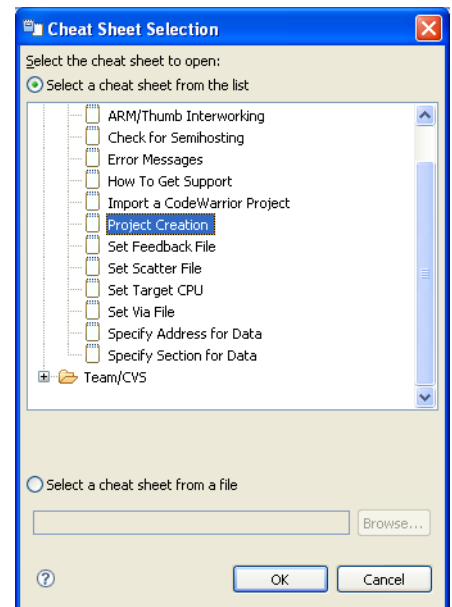
---

### 3.18 Using ARM-specific cheat sheets

Cheat sheets are working examples that you can use to guide you through a specific task. Each step in the task is listed in the Cheat Sheets view and the current step is highlighted and expanded. You must perform each step in turn to complete the task.

To use a cheat sheet:

1. Select **Cheat Sheet...** from the **Help** menu.
2. Select a cheat sheet from the list or use **Browse...** to select from a file, see the following figure.



**Figure 3-17 Selecting a cheat sheet**

3. Click on **OK**.
4. Click on the **Click to Begin** link on the Introduction step to start the task. If you open a cheat sheet more than once, the link on the Introduction step changes text to **Click to Restart**.
5. Follow the instructions step by step. When you complete an instruction, click on the **Click to Complete** link to move on to the next instruction. Some of the instructions might have a **Click to Perform** link for you to use if you want that instruction to be performed automatically by the cheat sheet.

The task is complete when you have performed every step listed in the Cheat Sheets view. This is shown by the tick icon shown in the left-hand margin of the cheat sheet see Figure 3-18.

Figure 3-18 shows a typical example of a cheat sheet. Steps one and two are shown as complete, step three (the current step) is highlighted and expanded ready for use. Four instructions are listed in step three, complete each instruction in turn. When step three is fully complete, the cheat sheet moves on to populate and reveal the instructions in step four.

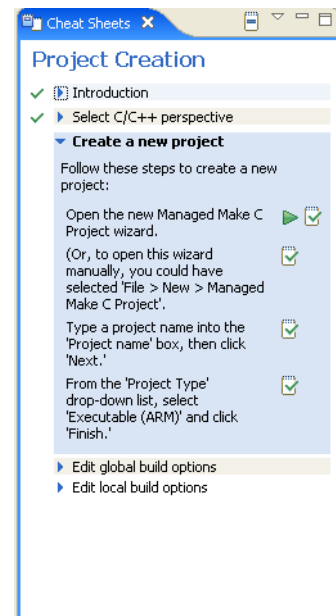
---

**Note**

---

Subsequent steps are not populated until you fully complete the previous step.

---



**Figure 3-18 Typical example of a cheat sheet**

## 3.19 Installing new features

To install new features:

1. Select **Install New Software...** from the **Help** menu.
2. Select a site from the **Work with** drop-down list or click **Add...** to add a new site.
3. Select the required plug-in check boxes. Click on **Next >**.
4. Review the list of plug-ins that you are about to install. Click on **Next >**.
5. Read each of the license agreements and accept them. If you do not accept a license agreement, you cannot install that feature. Click on **Finish**.
6. Click on **Yes** to restart Eclipse and complete the installation.

To install updated features, select **Check for Updates** from the **Help** menu.

### ———— **Note** ————

You can change the default setting for downloading and installing updated features by selecting **Install/Update** → **Automatic Updates** in the Preferences dialog box.

### 3.19.1 See also

#### **Reference**

- *Preferences dialog box* on page 3-24.

## 3.20 Restrictions of use

This section lists the specific restrictions and peculiarities that apply when using Eclipse.

### Organizing projects

The recommended structure for project source files is to create them in the project folder or sub-folder. If a source file is created in a folder that is higher than the project, an absolute link is created.

### Opening an existing Eclipse project

You must use the import wizard to import an existing Eclipse project into your workspace.

### Inter-project dependencies

Nested projects are not supported. Each project must be organized as a discrete entity. Inter-project dependencies can be set up by referencing other projects that reside in your workspace. Select **Project** → **Properties** → **Project References** from the main menu to manually add references.

**Link order** Specifying the link order of your object files within the same project is not possible with Eclipse. As a workaround, if you split your object files into different projects, you can specify the project build order. Select **Window** → **Preferences...** → **General** → **Workspace** → **Build Order** from the main menu.

### Restore Defaults

Restoring the defaults of a project discards all information that is not a part of the project type. All settings changed in the New Project Wizard are lost.



# Chapter 4

## Working with projects

You can use Eclipse to create projects for ARM® targets. Projects are top level folders in your workspace that contain related files and sub-folders. A project must exist in your workspace before a new file can be added or an existing file can be imported.

The following topics describe how to work with Eclipse projects:

### Tasks

- *Creating a new C or C++ project* on page 4-4
- *Importing an existing Eclipse project* on page 4-5
- *Accessing the build properties for a specific file or project* on page 4-8
- *Configuring the C/C++ build settings* on page 4-9
- *Updating a project to a new toolchain* on page 4-12
- *Adding a new source file to your project* on page 4-13.

### Concepts

- *Project types* on page 4-2.

## 4.1 Project types

Different project types are provided with Eclipse, depending on the requirements of your project.

---

**Note**

The list of available projects might vary depending on the installed plug-ins.

---

### 4.1.1 Makefile project

Use the **Makefile** project to create a project built by a Makefile that you can write yourself. Eclipse does not modify Makefile projects.

### 4.1.2 Executable

Use the **Executable** project type to create an executable ELF image.

### 4.1.3 Static library

Use the **Static Library** project type to build a library of ELF object format members.

---

**Note**

It is not possible to debug or run a stand-alone library file until it is linked into an image.

---

### 4.1.4 Shared Library

Use the **Shared Library** project type to build a dynamic library.

### 4.1.5 Build configurations

By default, the new project wizard provides two separate build configurations:

- |                |   |
|----------------|---|
| <b>Debug</b>   | The debug target is configured to build output binaries that are fully debuggable, at the expense of optimization. It configures the compiler optimization setting to minimum (level 0), to provide an ideal debug view for code development. |
| <b>Release</b> | The release target is configured to build output binaries that are highly optimized, at the expense of a poorer debug view. It configures the compiler optimization setting to high (level 3).  |

In all new projects, the Debug configuration is automatically set as the active configuration. This can be changed in the C/C++ Build Settings panel of the Project Properties dialog box.

#### 4.1.6 See also

##### Tasks

- *Accessing the dynamic help* on page 3-34
- *Creating a new C or C++ project* on page 4-4
- *Importing an existing Eclipse project* on page 4-5.

##### Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 4.2 Creating a new C or C++ project

To create a new C or C++ Project:

1. Select **File** → **New** → **Project...** from the main menu.
  2. Expand the **C/C++** group.
  3. Select either **C Project** or **C++ Project**.
  4. Click on **Next**.
  5. Enter a project name.
  6. Leave the **Use default location** option selected so that the project is created in the default folder shown. Alternatively, deselect this option and browse to your preferred project folder.
  7. Select the type of project that you want to create.
  8. Click on **Finish** to create your new project.
- The project is visible in the Project Explorer view.

### 4.2.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- *Accessing the build properties for a specific file or project* on page 4-8
- *Configuring the C/C++ build settings* on page 4-9
- *Adding a new source file to your project* on page 4-13.

#### Concepts

- *Project types* on page 4-2.

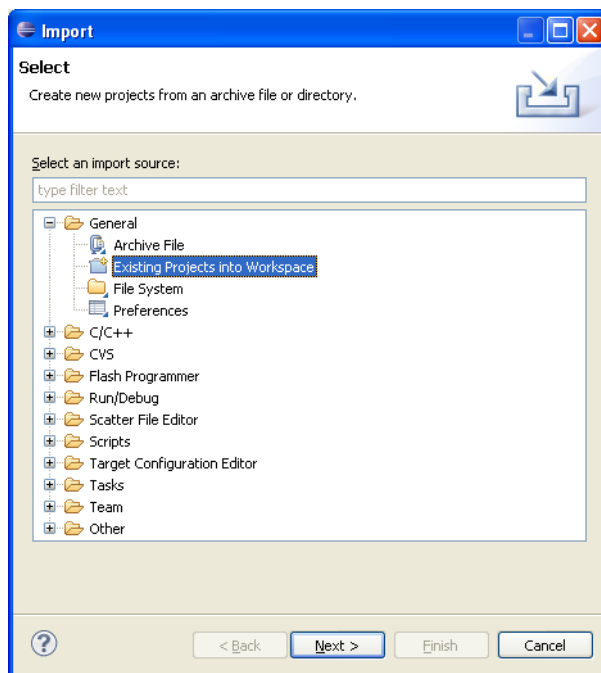
#### Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 4.3 Importing an existing Eclipse project

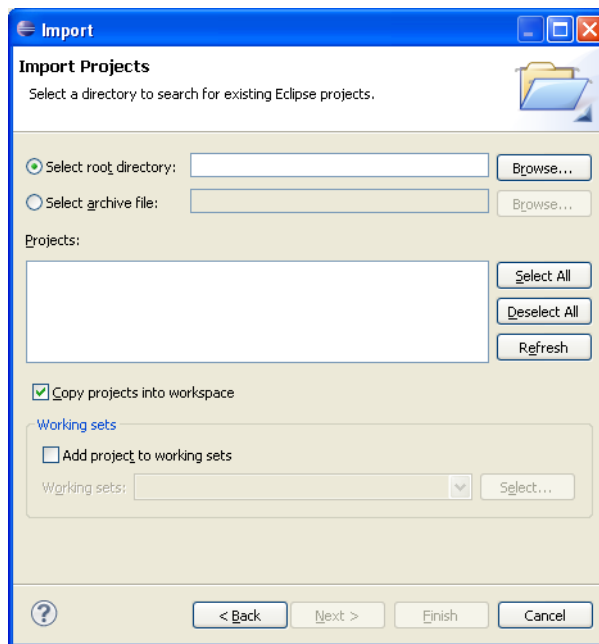
To import an existing Eclipse project into your workspace:

1. Select **Import...** from the **File** menu.
2. To import an existing project, select **Existing Project into Workspace**. Click on **Next**.



**Figure 4-1** Selecting the import source type

3. Click on **Browse** to select the import folder containing the project that you want to import.
4. In the Projects panel select the project that you want to import.
5. Select **Copy projects into workspace** if required or deselect to create links to your existing project(s) and associated files.



**Figure 4-2** Selecting an existing Eclipse projects for import

6. Click on **Finish**.

———— **Note** ————

If your existing project contains project settings from an older version of the build system, you are given the option to update your project. Using the latest version means that you can access all the latest toolchain features.

The imported project is visible in the Project Explorer view.

### 4.3.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- *Accessing the build properties for a specific file or project* on page 4-8
- *Configuring the C/C++ build settings* on page 4-9
- *Adding a new source file to your project* on page 4-13.

#### Concepts

- *Project types* on page 4-2.

## Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 4.4 Accessing the build properties for a specific file or project

The C/C++ Build configuration panels enable you to set up the compilation tools for a specific build configuration, in your project. The settings in these panels determine how the compilation tools build an ARM executable image or library.

---

### Note

---

Build configuration settings can be applied to individual files and complete projects.

---

To access the build configuration panels that affect all the source files in your project:

1. Select the source file or project in the Project Explorer view.
2. Select **Properties** from the **Project** menu.
3. Expand **C/C++ Build** in the Properties dialog box.
4. Select **Settings**.
5. The Configuration panel shows the active configuration. Click **Manage Configurations...** from the Configuration panel to create a new build configuration or change the active setting.
6. The compilation tools available for the current project, and their respective configuration panels are displayed in the **Tool Settings** tab.

### 4.4.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- *Configuring the C/C++ build settings* on page 4-9.

#### Concepts

- *Project types* on page 4-2.

#### Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.



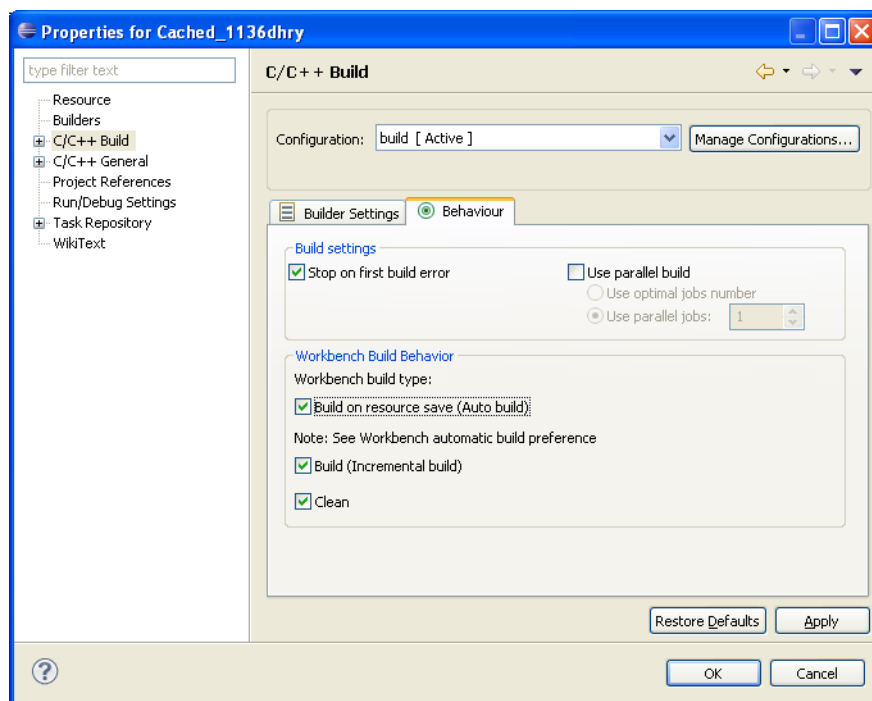
## 4.5 Configuring the C/C++ build settings

A build is the process of compiling and linking source files to generate an output file. A build can be applied to either a specific set of projects or the entire workspace. It is not possible to build an individual file or sub-folder.

Eclipse provides an incremental build that applies the selected build configuration to resources that have changed since the last build. Another type of build is the Clean build that applies the selected build configuration to all resources, discarding any previous build states.

### Automatic

This is an incremental build that operates over the entire workspace and can run automatically when a resource is saved. This setting must be enabled for each project by selecting **Build on resource save (Auto build)** in the **Behaviour** tab. By default, this behavior is not selected for any project.



**Figure 4-3 Workbench build behavior**

You must also ensure that **Build Automatically** is selected from the **Project** menu. By default, this menu option is selected.

## Manual

This is an incremental build that operates over the entire workspace on projects with **Build (Incremental build)** selected. By default, this behavior is selected for all projects.

You can run an incremental build by selecting **Build All** or **Build Project** from the **Project** menu.

---

### Note

---

Manual builds do not save before running so you must save all related files before selecting this option! To save automatically before building, you can change your default settings by selecting **Preferences...** → **General** → **Workspace** from the **Window** menu.

---

## Clean

This option discards any previous build states including object files and images from the selected projects. The next automatic or manual build after a clean, applies the selected build configuration to all resources.

You can run a clean build on either the entire workspace or specific projects by selecting **Clean...** from the **Project** menu. You must also ensure that **Clean** is selected in the **C/C++ Build** → **Behaviour** tab of the Preferences dialog box. By default, this behavior is selected for all projects.

Build order is a feature where inter-project dependencies are created and a specific build order is defined. For example, an image might require several object files to be built in a specific order. To do this, you must split your object files into separate smaller projects, reference them within a larger project to ensure they are built before the larger project. Build order can also be applied to the referenced projects.

### 4.5.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- *Creating a new C or C++ project* on page 4-4
- *Importing an existing Eclipse project* on page 4-5
- *Accessing the build properties for a specific file or project* on page 4-8.
- *Configuring the C/C++ build settings* on page 4-9.

#### Concepts

- *Project types* on page 4-2.

## Reference

- *Menus* on page 3-19
- *Toolbars* on page 3-20
- *Perspectives and views* on page 3-14
- *Preferences dialog box* on page 3-24
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 4.6 Updating a project to a new toolchain

If you have several products installed, only the latest toolchain is listed in the New Project wizard. Therefore, if you have projects that use an older toolchain, you must update them to the latest toolchain.

To update them to use the new toolchain:

1. Right-click on the project in the Project Explorer view.
2. Select **Convert To...** from the context menu to display the Project Converters dialog box.
3. Select the required toolchain.

———— **Note** ————

Beware that if you convert back to an earlier version you might lose toolchain functionality.

—————

4. Click **OK**.

### 4.6.1 See also

#### Tasks

- *Accessing the dynamic help* on page 3-34
- *Creating a new C or C++ project* on page 4-4
- *Importing an existing Eclipse project* on page 4-5.

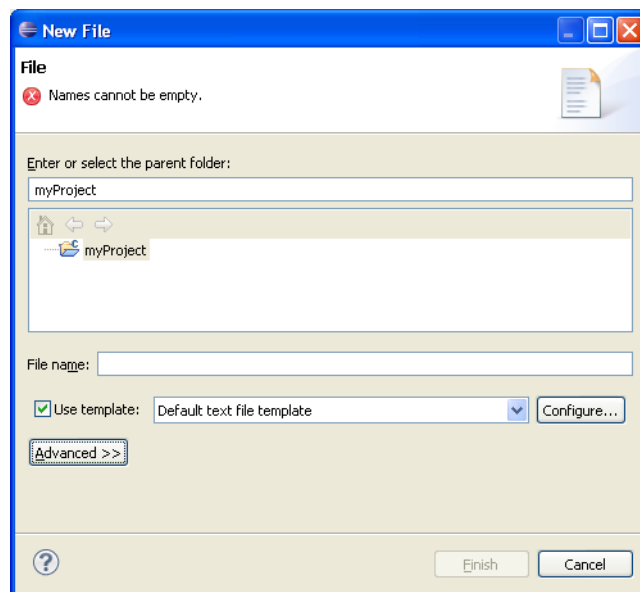
#### Concepts

- *Project types* on page 4-2.

## 4.7 Adding a new source file to your project

To add a new source file to your project:

1. Select **File** → **New** → **File from Template** from the main menu.
2. Select your project folder or sub-folder from the New File dialog box.



**Figure 4-4 Naming your file**

3. Enter a filename with the relevant extension in the **File name** field.
4. Accept the default template or click **Configure...** if you want to change it.
5. Click on **Finish**.

The new source file is visible in the Project Explorer view.

### **Note**

- You can also create files, or drag and drop files directly into the project folder, using the file system. To update the views in Eclipse, click on the relevant project in the Project Explorer view and select **Refresh** from the **File** menu.
- You can also drag and drop files directly into a project folder, in the Project Explorer view of Eclipse.



# Chapter 5

## Working with editors

The following topics describe how to use the editors when developing a project for an ARM® target:

### Concepts

- *Overview of the C/C++ editor on page 5-2*
- *Overview of the ARM assembler editor on page 5-3.*

## 5.1 Overview of the C/C++ editor

The standard C/C++ editor is provided by the CDT plug-in that provides C and C++ extensions to Eclipse. It provides syntax highlighting, formatting of code and content assistance when editing C/C++ code. See the *C/C++ Development User Guide* in the help system. Select **Help** → **Help Contents** from the main menu.

If this is not the default editor, right-click on a source file in the Project Explorer view and select **Open With** → **C/C++ Editor** from the context menu.



## 5.2 Overview of the ARM assembler editor

The ARM® assembler editor provides syntax highlighting, formatting of code and content assistance for labels in ARM assembly language source files. You can change the default settings in the Preferences dialog box.

If this is not the default editor, right-click on your source file in the Project Explorer view and select **Open With** → **ARM Assembler Editor** from the context menu.

The following shortcuts are also available for use:

**Table 5-1 ARM assembler editor shortcuts**

Content assist	Content assist provides auto-completion on labels existing in the active file. When entering a label for a branch instruction, Partially type the label and then use the keyboard shortcut Ctrl+Space to display a list of valid auto-complete options. Use the Arrow Keys to select the required label and press Enter to complete the term. Continue typing to ignore the auto-complete list.
Editor focus	The following options change the editor focus: <ul style="list-style-type: none"> <li>• Outline View provides a list of all areas and labels in the active file. Click on an area or label to move the focus of the editor to the position of the selected item.</li> <li>• Select a label from a branch instruction and press F3 to move the focus of the editor to the position of the selected label.</li> </ul>
Formatter activation	Press Ctrl+Shift+F to activate the formatter settings.
Block comments	Block comments are enabled or disabled by using Ctrl+Semicolon. Select a block of code and apply the keyboard shortcut to change the commenting state.

### 5.2.1 See also

#### Concepts

- *Preferences dialog box* on page 3-24.



# Chapter 6

## Working with Remote System Explorer

The following topics describe the Remote System Explorer and how you can use it to access resources on the host workstation and remote targets:

### Tasks

- *Adding Remote System Explorer views to the DS-5 Debug perspective* on page 6-3.

### Concepts

- *Overview of the Remote System Explorer* on page 6-2.

### Reference

- *Remote Systems view* on page 6-4
- *Remote System Details view* on page 6-5
- *Remote Scratchpad view* on page 6-6
- *Terminals view* on page 6-7.

## 6.1 Overview of the Remote System Explorer

The *Remote System Explorer* (RSE) enables you to:

- set up Linux SSH connections to remote targets using TCP/IP
- create, copy, delete, and rename resources
- set the read, write, and execute permissions for resources
- edit files by double-clicking to open in the C/C++ editor
- execute commands on the remote target
- view and kill running processes
- transfer files between the host workstation and remote targets
- launch terminal views.

Useful RSE views that you can add to the DS-5 Debug perspective are:

- Remote Systems
- Remote System Details
- Remote Scratchpad
- Terminals.

### 6.1.1 See also

#### Tasks

- *Adding Remote System Explorer views to the DS-5 Debug perspective* on page 6-3.

#### Reference

- *Remote Systems view* on page 6-4
- *Remote System Details view* on page 6-5
- *Remote Scratchpad view* on page 6-6
- *Terminals view* on page 6-7
- *Miscellaneous icons* on page 7-10.

## 6.2 Adding Remote System Explorer views to the DS-5 Debug perspective

To add a *Remote System Explorer* (RSE) view to your DS-5 Debug perspective:

1. Ensure that you are in the DS-5 perspective. To can change perspective either:
  - use the perspective toolbar
  - select select **Window** → **Open perspective** → **DS-5 Debug** from the main menu.
2. Select **Window** → **Show View** → **Other...** to open the Show View dialog box.
3. Select the required view from the **Remote Systems** group.
4. Click **OK**.

### 6.2.1 See also

#### Concepts

- *Overview of the Remote System Explorer* on page 6-2.

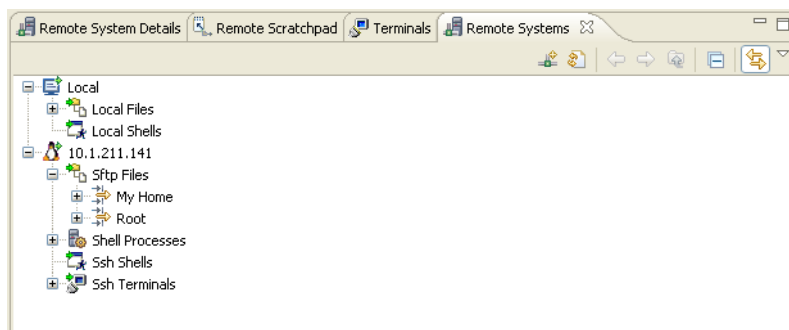
#### Reference

- *Remote Systems view* on page 6-4
- *Remote System Details view* on page 6-5
- *Remote Scratchpad view* on page 6-6
- *Terminals view* on page 6-7
- *Miscellaneous icons* on page 7-10.

## 6.3 Remote Systems view

This hierarchical tree view enables you to:

- set up a Linux connection to a remote target using the *Secure SHell* (SSH) protocol
- access resources on the host workstation and remote targets
- display a selected file in the C/C++ editor
- open the Remote System Details view and show the selected connection configuration details in a table
- open the Remote Monitor view and show the selected connection configuration details
- import and export the selected connection configuration details
- connect to the selected target
- delete all passwords for the selected connection
- open the Properties dialog box and display the current connection details for the selected target.



**Figure 6-1 Remote Systems view**

### 6.3.1 See also

#### Concepts

- *Overview of the Remote System Explorer* on page 6-2.

#### Reference

- *Remote System Details view* on page 6-5
- *Remote Scratchpad view* on page 6-6
- *Terminals view* on page 6-7
- *Miscellaneous icons* on page 7-10.

## 6.4 Remote System Details view

This tabular view enables you to:

- set up a Linux connection to a remote target using the *Secure SHell* (SSH) protocol
- access resources on the host workstation and remote targets
- display a selected file in the C/C++ editor
- open the Remote Systems view and show the selected connection configuration details in a hierarchical tree
- open the Remote Monitor view and show the selected connection configuration details
- import and export the selected connection configuration details
- connect to the selected target
- delete all passwords for the selected connection
- open the Properties dialog box and display the current connection details for the selected target.

Resource	Remote system type	Connection status	Host name	Description
Local	Local	Some subsystems connected	LOCALHOST	
10.1.211.141	Linux	Some subsystems connected	10.1.211.141	

**Figure 6-2 Remote System Details view**

### 6.4.1 See also

#### Concepts

- *Overview of the Remote System Explorer on page 6-2.*

#### Reference

- *Remote Systems view on page 6-4*
- *Remote Scratchpad view on page 6-6*
- *Terminals view on page 6-7*
- *Miscellaneous icons on page 7-10.*

## 6.5 Remote Scratchpad view

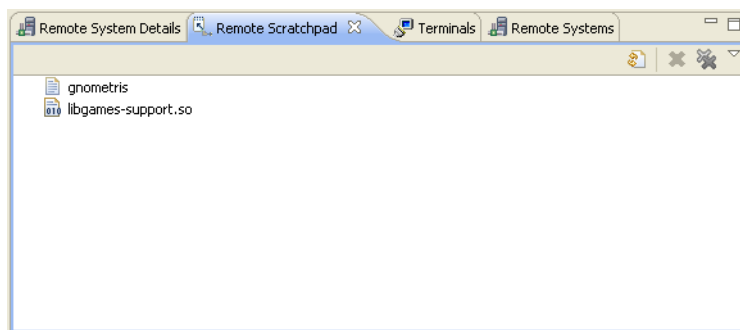
The Remote Scratchpad view is an electronic clipboard where you can copy and paste or drag and drop useful files and folders into this view for use at a later point in time. This enables you to keep a list of resources from any connection in one place.

---

### Note

Be aware that although the scratchpad only shows links, any changes made to a linked resource also changes it in the original file system.

---



**Figure 6-3 Remote Scratchpad**

### 6.5.1 See also

#### Concepts

- *Overview of the Remote System Explorer* on page 6-2.

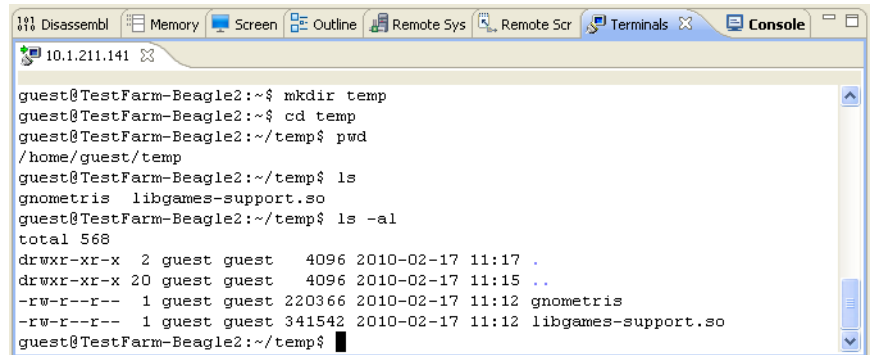
#### Reference

- *Remote Systems view* on page 6-4
- *Remote System Details view* on page 6-5
- *Terminals view* on page 6-7
- *Miscellaneous icons* on page 7-10.



## 6.6 Terminals view

The Terminals view enables you to enter shell commands directly on the target without launching any external application. For example you can browse remote files and folders by entering the `ls` or `pwd` commands in the same way as you would on a Linux terminal.



**Figure 6-4** Terminals view

### 6.6.1 See also

#### Concepts

- *Overview of the Remote System Explorer* on page 6-2.

#### Reference

- *Remote Systems view* on page 6-4
- *Remote System Details view* on page 6-5
- *Remote Scratchpad view* on page 6-6
- *Miscellaneous icons* on page 7-10.



# Chapter 7

## Terminology, shortcuts and icons

This following topics describe some of the terminology used in *Eclipse for DS-5 Using Eclipse*, useful keyboard shortcuts, and menu and toolbar icons:

### Reference

- *Terminology* on page 7-2
- *Keyboard shortcuts* on page 7-4
- *Menu and toolbar icons* on page 7-6.

## 7.1 Terminology

The following terminology is used in this document:

<b>Block</b>	A small sub-division of a flash device that can be programmed.
<b>Device</b>	A component on a target containing the application that you want to debug.
<b>Dialog box</b>	A small page containing tabs, panels and editable fields prompting you to enter information.
<b>Editor</b>	A view that controls the visual aspects of source code for a specific file type.
<b>Erase</b>	A feature of a flash device where memory cells are reset to a known value.
<b>Flash device</b>	A set of flash memory that has a single command interface.
<b>Panel</b>	A small area in a dialog box or tab to group editable fields.
<b>Perspective</b>	A page within the Eclipse window containing a set of related views, editors, menus, and toolbars.
<b>Program</b>	A term used to describe the storing of data on a flash device.
<b>Project</b>	A group of related files and folders in Eclipse.
<b>Resource</b>	A generic term used to describe a project, file, folder, or a combination of these.
<b>Send To</b>	A term used to describe sending a file to a target.
<b>Tab</b>	A small overlay page containing panels and editable fields within a dialog box to group related information. Clicking on a tab brings it to the top.
<b>Target</b>	A location to send a file, for example, a flash image.
<b>View</b>	A small page to display related information for a specific function.
<b>Width</b>	The smallest number of bits (8, 16 or 32) that can be natively accessed by a flash device.
<b>Wizard</b>	A group of dialog boxes to guide you through a common tasks, for example, creating new files and projects.
<b>Workbench</b>	A window containing perspectives, menus, and toolbars.

**Workspace** An area designated on your file system to store files and folders related to your projects.

## 7.2 Keyboard shortcuts

The most common keyboard shortcuts available for use with Eclipse are:

**F3** Click on an assembler label from a branch instruction or a C/C++ calling function and press F3 to move the editor focus to the position of the selected item.

**F10** Use in conjunction with the arrow keys to access the main menu.

**Alt+F4** Exit Eclipse.

**Alt+Left arrow**  
Go back in navigation history.

**Alt+Right arrow**  
Go forward in navigation history.

**Ctrl+Semicolon**  
Provided with the ARM® assembler editor to add comment markers to a selected block of code in the active file.

**Ctrl+End** Moves the editor focus to the end of the code.

**Ctrl+B** Build all projects in the workspace that have changed since the last build.

**Ctrl+Home** Moves the editor focus to the beginning of the code.

**Ctrl+F** Opens the Find or Find/Replace dialog box to search through the code in the active editor. Some editors are read-only and therefore disable this functionality.

**Ctrl+F4** Close the active file in the editor view.

**Ctrl+F6** Cycles through open files in the editor view.

**Ctrl+F7** Cycles through available views.

**Ctrl+F8** Cycles through available perspectives.

**Ctrl+F10** Use in conjunction with the arrow keys to access the drop-down menu.

**Ctrl+L** Move to a specified line in the active file.

**Ctrl+Q** Move to the last edited position in the active file.

**Ctrl+Space** Provides auto-completion on selected functions in editors.

**Shift+F10** use in conjunction with the arrow keys to access the context menu.

**Ctrl+Shift+F**

Activates the code style settings in the Preferences dialog box and apply them to the active file.

**Ctrl+Shift+L**

Opens a small page with a list of all keyboard shortcuts.

**Ctrl+Shift+R**

Opens the Open resource dialog box.

**Ctrl+Shift+T**

Opens the Open Type dialog box.

**Ctrl+Shift+/**

Provided with the C/C++ editor to add comment markers to the start and end of a selected block of code in the active file.

## 7.3 Menu and toolbar icons




These tables list the most common menu and toolbar icons available for use with Eclipse. For information on icons, markers, and buttons not listed in the following tables, see the standard *Workbench User Guide* or the *C/C++ Development User Guide* in the Help Contents.

If you leave the mouse pointer positioned on a toolbar icon for a few seconds without clicking, a tooltip appears informing you of the purpose of the icon.









### 7.3.1 Perspective icons

**Table 7-1 Perspective icons**

Button	Description	Button	Description
	Open new perspective		C/C++
	Fast view bar		







### 7.3.2 View icons

**Table 7-2 View icons**

Button	Description	Button	Description
	Display drop-down menu		Synchronize view contents
	Minimize		Maximize
	Restore		Close









### 7.3.3 View markers

**Table 7-3 View markers**

Button	Description	Button	Description
	Bookmark		Information
	Task		Search result
	Error		Warning



7.3.4 Editor icons

Table 7-4 Editor icons

Button	Description	Button	Description
	Save the active file		Save all files
	Print the active file		Close view
	Create new configuration		Duplicate selected configuration
	Delete selected configuration		Collapse configuration tree

7.3.5 Configuration icons

Table 7-5 Configuration icons

Button	Description	Button	Description
	Open Run Configurations dialog box		Open Debug Configurations dialog box

7.3.6 Outline icons

Table 7-6 Outline icons





































Button	Description	Button	Description
	Hide fields		Hide static members
	Hide non-public members		Sort alphabetically
	Class		Namespace
	Macro definition		Enum
	Enumerator		Variable

Table 7-6 Outline icons (continued)

Button	Description	Button	Description
	Protected field		Private field
	Public field		Include
	Protected method		Private method
	Public method		Struct
	Type definition		Union
	Function		Target configuration file
	Include file		Memory
	Peripheral		Register
	Bitfield		Enumeration definition
	Map rules		Tools information








7.3.7 Miscellaneous icons

Table 7-7 Miscellaneous icons

Button	Description	Button	Description
	Open a new resource wizard		Open new project wizard
	Open new folder wizard		Open new file wizard
	open search dialog box		Display context-sensitive help








7.3.8 Navigation icons

Table 7-8 Navigation icons

Button	Description	Button	Description
	Navigate back		Navigate forwards
	Navigate up one level		Print active page
	Open help instruction page		Synchronize TOC with active page
	Bookmark active page		

7.3.9 Help Contents icons

Table 7-9 Help Contents icons

Button	Description	Button	Description
	Display list of all documents		Display list of documents in last search
	Display list of context-sensitive help links to related topics		Display list of all bookmarks
	Maximize frame		Restore frame
	Synchronize TOC with active page		

