Tutorial Introduction

PURPOSE:

- To explain how CodeWarrior organizes files, libraries, and tool settings into a document known as a "project"

OBJECTIVES:

- Describe a project.
- Describe a group.
- Explain how to create a project file.
- Describe how to manage files within a project.

CONTENT:

- 17 pages
- 3 questions

LEARNING TIME:

- 15-20 minutes

In this tutorial, we'll discuss how CodeWarrior manages the files that are used to build a program. These files and related information, such as compiler and linker settings, are stored in a file known as a "project."

Upon completion of this tutorial, you'll be able to describe a project and a group. You'll also be able to create a project and describe how to manage files within a project.

Creating a Project From Stationery

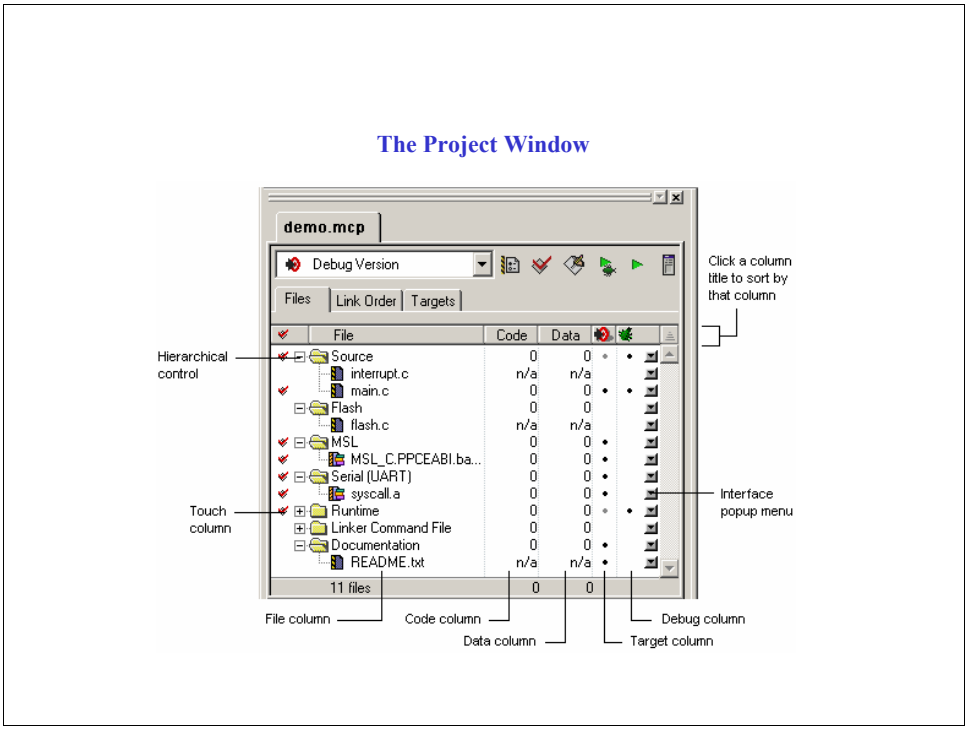
- The term “Stationery” means a template for a new project.
- Stationery provides everything needed for a read-to-run project
  - Basic “hello world”-type source code
  - Necessary Standard Libraries
  - Compiler, linker, debugger settings
  - Target Interface configuration files
- Stationery Wizards
  - Board
  - Processor
  - Language
  - Connection Method

A CodeWarrior project is a “container” for all of your source code files and libraries and for your compiler/linker settings. This information is used to create a program.

A stationery project is simply a standard CodeWarrior project file that serves as a "template" for new projects you create. CodeWarrior makes a duplicate of this stationery file for use in your project.

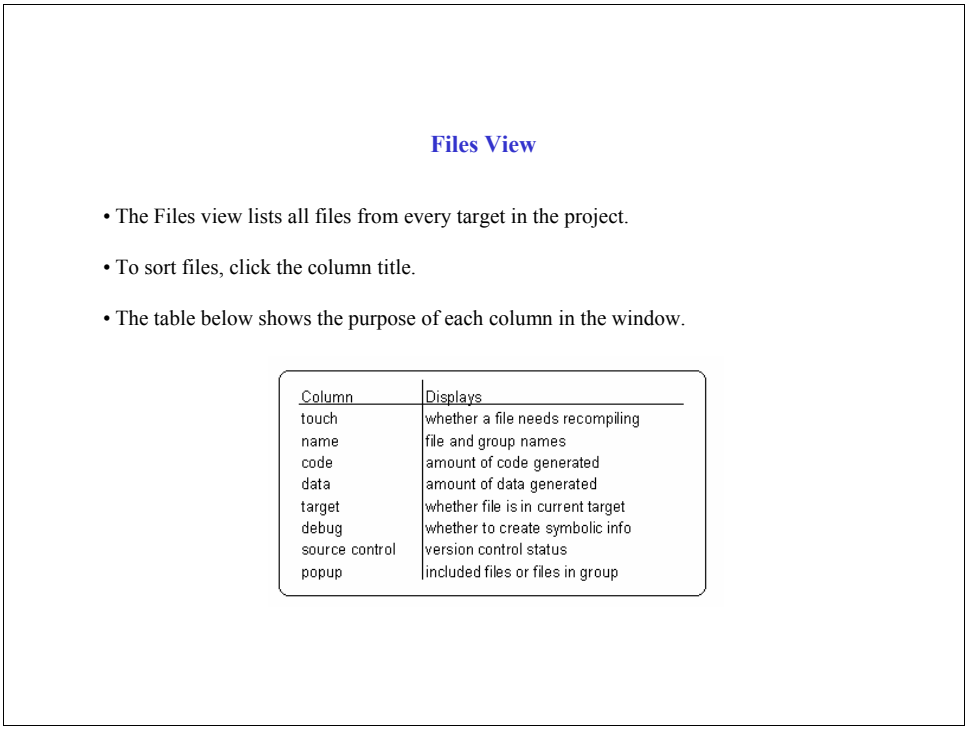
To create a new project using stationery, select the File ==> New command sequence. A dialog box will appear with a list of available stationery types. Select from the list of stationery types, and wizards. Select a name and location for your new project, and then Click OK. A New Project dialog box will appear, from which choose the stationery project, then click OK. All files will be copied, paths set, and settings specified.

Stationery Wizards guide the new project creation, by asking board type, processor, language, and connection method. Then tailoring the resulting project to fit those needs.



Next, let's look at the Project window, which is the main information window in CodeWarrior. The Project window portrays information for a CodeWarrior project. You can have one project open, or multiple projects open simultaneously.

The tabs at the top of the window are used to select different projects. There is only one project open, demo.mcp. The other tabs are used to select different views for the project. The Files view lists all files and libraries for every Build target in the project. The Link Order view shows you the order everything is linked for the active Build target. You are able to change that order, a capability that is important for some platforms. The Targets view enables you to change the current Build target, create/delete targets, and to create target dependencies.



The Project window's Files view lists *all* files, even if they are *NOT* part of the currently active Build target whose name is displayed in the Project window tool bar.

Files belong to Build targets, not to projects. Removing a file from the Files view means you are removing it from the project, the equivalent of removing the file from all build targets. Methods for adding or removing a file from a single Build target are discussed in another module of this course.

The source-control column only appears when you are using Version Control System (VCS) software with your project.

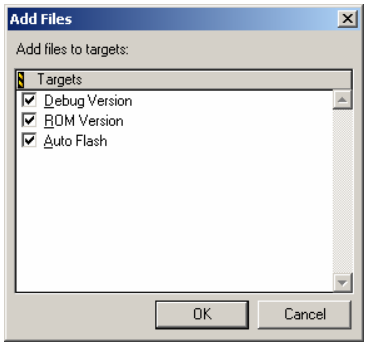
Using the File Context Menu

- Right-click a file name. A menu will appear.
- Choose a command from the menu.
- Commands available are as follows:
  - Open in Windows Explorer
  - file commands from the Project menu
  - Add Files…
  - Add Group…
  - Remove

Right-clicking a file in the Project window is a good shortcut to access common commands that are used to work with a particular file, such as removing the file, pre-compiling or disassembling the file.

When the Project window is in the Targets view, right-clicking a target name portrays target-related commands, such as the option to Make, or Delete that target.

Adding a File to a Project



- Menu commands --
  - Project ==> Add Files
    -- a standard add file dialog box appears
  - Project ==> Add *window name* to Project
    -- a Save As dialog box appears
- Added files appear at the current selection.
- Dragging a file or directory from the desktop -
  - adds an entire directory structure.
  - files appear where dropped.
- With multiple targets, you will be asked which to use, as the dialog box at the left shows.

If you add a file to a project using menu commands, the Project Manager positions the added file relative to the first selected item in the Files view. If the first selected item is a file, the added file appears before the selected item. If the first selected item is a group, and if that group has no selected files, the added file appears at the end of, but within, the group. If there's nothing selected, the added file appears at the end of the file list in the Files view.

Dragging multiple files, or even an entire directory, is the fastest way to add files to a project. If you drag a directory to the project window, CodeWarrior creates a file layout that mirrors the directory layout.

If your project contains more than one Build target, for example a Debug build target, and a Flash build target, CodeWarrior prompts you to identify the target to which you wish to add the file. You can add the file to one or more targets as appropriate. If the project contains only one Build target, the files are automatically added to that target.

Removing a File From a Project

- To remove a file, select the File(s) in the Files view of the Project Manager.
- Use the Edit ==> Remove command sequence.
- This action removes the File(s) from all Build targets.

The fastest way to remove a file is to use the Context menu. You can remove more than one file at a time by selecting all of the files you want to remove using your platform's keyboard modifiers, then right-click one of the selected files, then select Delete.

Please note that deleting a file from the project doesn't delete the file from your computer's hard drive. It simply removes the reference to that file from the project window. The file on the disk remains unchanged.

Creating a Group

- To create a group, drop a directory onto the Files view.
  - The entire desktop hierarchy is replicated.
  - The new group is assigned the directory name.
  - All folders contained become subgroups.
  - All files contained are added to the project.

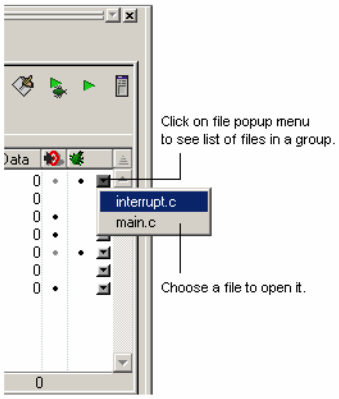
- Or, use the Project ==> Create Group… command sequence.
  - You must assign a name to the group. Use the dialog box provided.
  - The group remains empty until files are assigned.

Next, let's discuss groups. Groups provide a way to organize project files in a logical manner. Groups are not synonymous with actual file directories. However, you can mirror a directory's structure using groups if you wish. The way you organize your project is strictly up to you. There are no hard and fast rules.

To create a new group, choose the Project ==> Create Group… sequence. You can then add files to this group manually or by dragging files to the new group. If you drag a directory to the project window, the Project Manager creates a new group using the directory name as the group name and automatically adds all files to that directory.

For most platform targets, groups only appear in the Files view of the Project window. For some targets, groups appear in the Link Order view as well.

Working With Groups



- Use the menu to open a file inside of a closed group.
- Double-click to rename a group.
- Drag the group to move it to another location.
- Hierarchical Open and Close controls are provided.
- Drag files and groups in or out.
  - Nested groups are supported in the Files view.

Working with groups is similar to working with files. You can move groups around by dragging them. You can nest groups as subgroups. If you need to rename a group for any reason, just double-click the group name and type a new name in the dialog box.

To examine a closed group, use the hierarchy controls to open it. If you want to see what files are in a group when the group is closed or open a file from a closed group, use the file menu, as shown on this page.

Removing a Group

- To remove a group, first select that group.
- Then, select the Edit ==> Remove command sequence.
- This action removes all files in the group.
- Files are removed from all Build targets in the project.

To remove a group, select the group and choose Delete from the Edit menu. Or, right-click the group and use the contextual menu. If you want to remove a group but preserve its contents, drag the files out of the group before removing the group.

Setting the Default Project

- Scenario --
  - An open source code file is in multiple open projects.
  - Identify the options that control a compile command.
- Settings that control compile commands --
  - the project from which the file was opened, or
  - the default project, in ambiguous situations
  - for example, the original project is closed
- The default project is either --
  - the first project opened, or
  - the project specified by the Project ==> Set Default Project command
- Activating a project window does not make it the default condition.

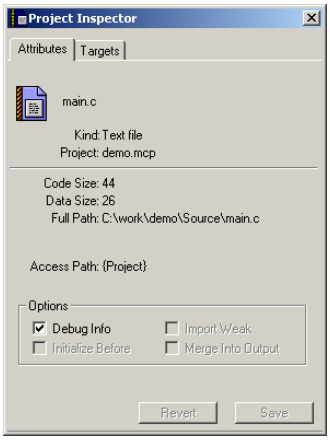
Let's look at other ways to manage files within a project, beginning with setting the default project.

The default project is typically the first project you open. It's the focus of any commands that you order the IDE to execute. You can change this focus to a different project by using the Set Default Project command in the Project menu. Or you can use that menu item to see what Project is presently the Default Project.

The default project setting only matters if the IDE encounters an ambiguous situation where it is unclear which settings control a file. For example, an open file exists that is used in multiple open projects, but the original project from which that file was opened is subsequently closed. If you then issue a Compile command for that file, CodeWarrior will use the settings in the default project to compile the file.

If you close the default project when multiple projects are open, the project whose window is closest to the top of the queue becomes the default project.

The Project Inspector Window



- Open this window using the View ==> Project Inspector command sequence.
- The Project Inspector window has two views:
  - attributes, detailed information about the file
  - targets, displays targets that include this file
- Most information and options are available in the one of the two Project Inspector Window views.
- The Project Inspector is useful to
  - view the full path for any file
  - view the access path for any file
  - view and set targets using a file

The Project Inspector window is used to view and change different bits of information in one or more files. Most of these settings can be viewed or changed in the Project window itself, although for only one file or group of files at a time. Using the Project Inspector window, you can change these settings for many files at once, even across different groups.

If you do not select a file or group, the Project Inspector window will open showing no information. You can then select a file or group to inspect.

The Project Inspector Window enables you to see the attributes of a file, the complete directory path for any file, the equivalent access path used to access the file of interest, and the Build targets that use this file.

Creating Custom Stationary

- To create custom stationery, first create a project.
- Then, set up the project in any manner you like.
  - add default files or placeholders
  - add libraries
  - specify target settings
- Next, move the project folder to the CodeWarrior Stationery folder.
- Use caution when working with absolute paths in stationery.

When possible, create new projects using one of the stationery projects provided. Creating custom stationary projects saves time and helps you to prevent mistakes. Modify your project to match your ideal starter project. Make whatever changes you need, for example, add different or additional libraries, core source files, specific target settings, and targets.

Once the project is created, compile the project to make sure everything works. Close the project and move the directory for the project to the CodeWarrior Stationery directory. Delete the data folder for the stationery project; it is no longer required.

Please note that the Project Manager keeps track of files by remembering the directories in which the files are located. These are called access paths to be discussed later. If you plan on creating custom stationery, please review the access path information first. Avoid absolute access paths, because they are a precise path from the root directory to the directory with the files. If your stationery includes absolute access paths, they will not work because the new project will probably be in a different location or on a system with a different directory tree.

Question

Which of the following statements best describes a project? Click on your choice.

A) A makefile that builds a program

B) A file that stores the compiler/linker settings needed to build a program

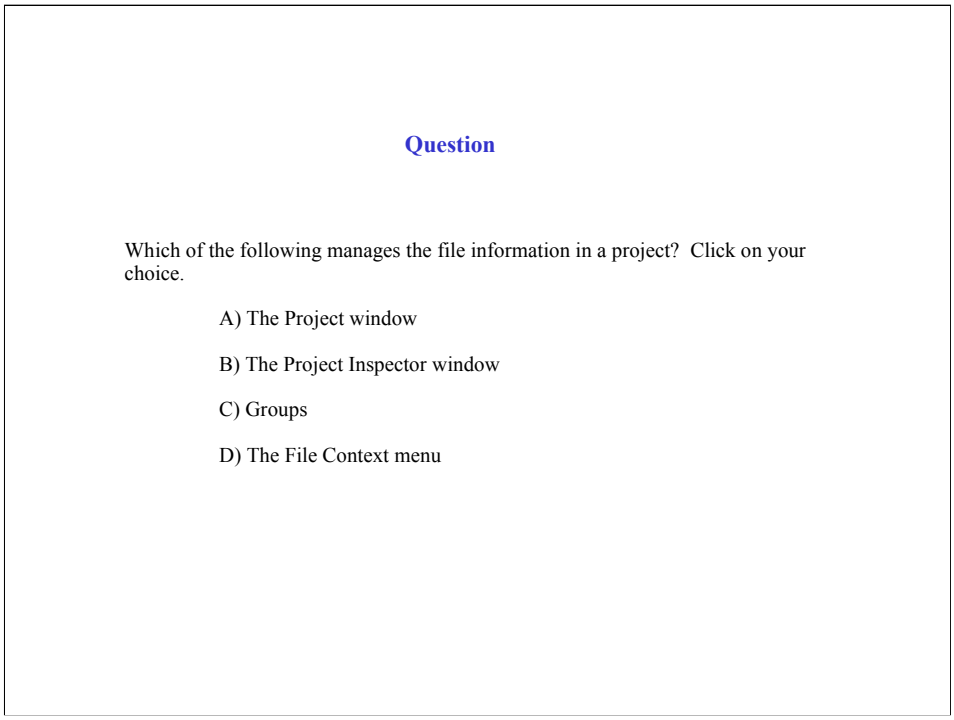
C) A file that stores all of the information necessary to build a program

D) A container that stores the files used to build the program

Let's complete this tutorial with a few questions.

Which of the following statements best describes a project?

Answer:
The project file contains all the information needed to build a program.



Which of the following manages the file information in a project?

Answer:

The Project window is the control center where you manage all of a project's files.

Question

Which of the following statements best describes a Group? Click on your choice.

A) A directory of files on the computer

B) A set of files organized by type

C) A way to logically organize files

Which of the following statements best describes a group?

Answer:

A Group is used to organize files in an order useful to you.

Tutorial Completion

- A project file contains all of the file names, libraries, settings and options needed to build a program in CodeWarrior.
- You can create a project using the File ==> New command.
  - There are stationary files that serve as templates with default libraries and settings for your use.
  - The Stationery Wizard will customize the template based on user feedback.
  - You can create a custom stationary project.
- The Project window is used to manage a project's files.
  - You can drag and drop to add files to the project.
- The Group object helps organize files in a logical manner.

In this tutorial, you've learned the definition of a project file. You've learned that most information related to a project file is controlled using the Project window. You've also learned about the available tools that help you manage the project's files, such as the File Context menu and the Project Inspector window.