

SoftBench to HP-UX Development Environment for Eclipse Migration Guide

This guide provides information on migrating from SoftBench Integrated Development Environment to HP-UX Development Environment for Eclipse, SoftBench features that are available, and their usage in HP-UX Development Environment for Eclipse.



Legal notices

© Copyright 2010 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Acknowledgments

Microsoft®, Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

Intel®, Pentium®, Itanium®, Intel Inside®, and the Intel Inside logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux® is a U.S. registered trademark of Linus Torvalds.

Eclipse® is a trademark of Eclipse Foundation Inc.

Table of Contents

Table of Contents	3
1 Introduction	5
Eclipse	5
HP-UX Development Environment for Eclipse	5
Features of HP-UX DEE	5
Migration Considerations	6
Installing HP-UX DEE	6
2 Using HP-UX DEE	7
Managing Tasks in SoftBench and HP-UX DEE	9
Setting up a Project	9
Creating a Project Using Project Build	9
Creating a Project Using External Build	10
Cloning a Project from an Existing Project	10
Repartitioning a project into two projects	11
Modifying a Project Definition	12
Creating Files within a Project	16
Source File	18
Header File	19
Adding Existing Files to a Project	19
Defining Targets	20
Building Projects and Targets	20
Setting the Compile Mode	21
Previewing the Build	21
Compiling Instead of Building	22
Interpreting Error Messages	22
Running the Build on a Remote System	23
Running Tools	23
Customizing your Environment	23
Adding and Removing Tool Icons	24
Choosing Tool Preferences	24
Accessing Distributed Data and Tools	24
Exiting the Workspace	24
Restoring your Previous Session	24
Getting Help	25
Repositories	25
3 Using Editors	26
Configuring the Editor	28
Editing Multiple Files	28
Managing Files and Projects Using the Editor	30
4 Using Code Advisor	31
5 Using Debugger	32
Stepping through a Program	33
Interrupting a Running Program	33
Interacting with a Program	34
Examining and Modifying Data in your Program	34
Breakpoints	35
WatchPoints	36
Tracing Program Flow	36

Handling Signals and Events	36
Correcting errors	36
Viewing Assembly Language and CPU Registers	37
Debugging Threaded Applications.....	37
Attaching a Debugger to a Running Program	37
Customizing the Debug Perspective.....	37
6 Using the Static Analyzer	38
Search Dialog	38
Search View	40
Context Menu options of the C/C++ Editor	42
7 Using File Compare.....	43
Comparing Files	43
Merging Files	44
8 Unsupported features.....	45
9 Support and Other Resources	46
Intended Audience	46
Conventions	46
References	46
Index.....	47

1 Introduction

Eclipse

Eclipse is an open source product managed by Eclipse.org. It is an Integrated Development Environment (IDE) that provides a multi-language software development environment. Eclipse IDE is written in Java and comprises of an extensible plug-in system. In its basic form, Eclipse consists of the Java Development Tools (JDT) and can be used to develop applications in Java. The extensible plug-in system of Eclipse enables users to extend the capabilities of Eclipse and develop applications in other languages such as C, C++, COBOL, Perl and so on. To develop applications in other languages, users will need to install plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages. For example, HP-UX Development Environment for Eclipse is a plug-in for Eclipse that enables users to develop remote C/C++ applications.

For more information about Eclipse, see:

<http://www.eclipse.org/>

HP-UX Development Environment for Eclipse

HP-UX Development Environment for Eclipse (HP-UX DEE) is a plug-in for the Eclipse IDE, and it inherits all the features of the Eclipse IDE. It enables you to develop C/C++ projects on the HP Integrity systems (Itanium IA) and the HP 9000 server (PA-RISC) running HP-UX 11i releases, using the Eclipse IDE on a Windows or Linux system. Unlike SoftBench, HP-UX DEE is installed on a local Windows or a Linux system.

HP-UX DEE enables you to edit your code, compile, run, debug and perform static analysis of your code. The HP-UX server provides the processing power to build, run, perform static analysis, and debug C/C++ programs.

HP-UX DEE accesses the project and the files on the remote system as local files using the mapped network drive. As a result, you can develop your C/C++ project remotely with the feel of developing the project locally.

HP-UX DEE uses Secure Shell (SSH) or Remote Shell (RSH) protocols to connect to the remote system. The SSH protocol provides a secure channel for transfer of data from the remote system to the local system where HP-UX DEE is running.

The following are the advantages of migrating from SoftBench to HP-UX DEE:

- HP-UX DEE enables you to develop C/C++ projects remotely from a Windows or a Linux system with the feel of local development.
- HP-UX DEE is integrated with the HP Code Advisor (cadvice), which is a powerful static analysis tool than that provided by SoftBench.
- HP-UX DEE is a free product but not an open source product.
- The HP-UX DEE webpage provides information regarding the technical support available for HP-UX DEE: www.hp.com/go/hpuxdee

Features of HP-UX DEE

You can use HP-UX DEE to:

- Create remote C/C++ projects using the Eclipse IDE on a Windows or Linux system.
- Invoke remote build on the project.
- Run and debug the remote project using the remote launch configuration.
- Perform memory debugging.
- Perform static analysis of the C/C++ code using the HP Code Advisor (cadvice), which is integrated with HP-UX DEE.

HP-UX DEE inherits the following feature from Eclipse:

- Views: A visual component in a workbench page. It provides alternative representations and ways to navigate a hierarchy of information in the Workbench. Eclipse includes views, such as Project Explorer view, Console view, Problems view, Outline view, and so on.

Migration Considerations

Before migrating a project from SoftBench to HP-UX DEE, ensure that you have the following.

- Host Workstation: HP-UX DEE supports developing projects on HP Integrity systems (Itanium) and the HP 9000 server (PA-RISC) running HP-UX 11iv2 and later releases. For information about the file systems required, the protocols used, and other software to be installed, see the *Installation Guide*.
- Remote System Connectivity: to connect to the remote system, you need to create a Network drive. For information about creating a network drive, see the *Installation Guide*.
- Language support: HP-UX DEE supports C and C++ programming languages.
- Makefile support: the Makefile generated by SoftBench is compatible for use in HP-UX DEE.

Installing HP-UX DEE

For information on how to install HP-UX DEE, see the *Installation Guide* at:

http://h21007.www2.hp.com/portal/download/files/unprot/hpc/eclipse-plugin/4_0_0/RDP%20Installation%20Guide.pdf

2 Using HP-UX DEE

The HP-UX DEE plug-in uses the same GUI as Eclipse.

Figure 2-1 shows the Eclipse Workbench. The top right corner shows the selected perspective (C/C++).

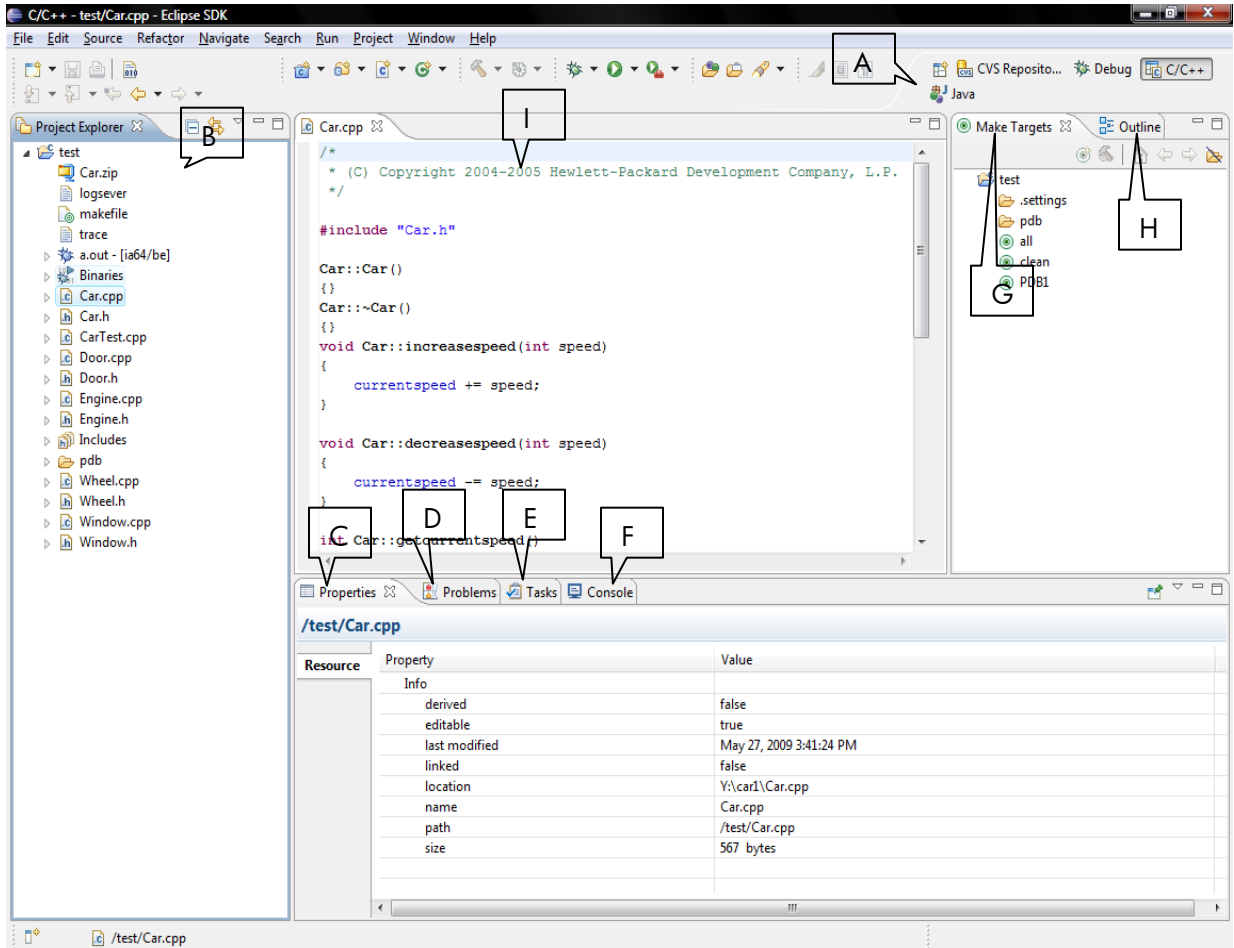


Fig 2-1: C/C++ Perspective

Table 2-1 provides a description of the Eclipse Workbench shown in Fig 2-1.

Table 2 -1 Eclipse Workbench Description

Label	Description
A	This area of the workbench shows the selected perspective.
B	Project explorer view: displays the projects that are open in the workspace.
C	Properties view: displays the property name and basic properties of a selected resource.
D	Problems view: displays the error/warning messages of the projects that are open in the workspace.
E	Tasks view: Eclipse enables you to track tasks by adding the tasks manually. These tasks are displayed in the Tasks view. For information on how to use the Tasks view, navigate to the following pages in the online help page for Eclipse: Workbench User Guide → Reference → User interface information → Views and Editors → Tasks view. Workbench User Guide → Concepts → Views → Tasks view.
F	Console view: displays the output of build actions and the output of the project.

Table 2-1 Eclipse Workbench Description

Label	Description
G	Make Targets view: enables you to create or select the Make targets for your project.
H	Outline view: displays the outline of a structured file that is currently open in the editor area. Its contents are editor-specific. To view the outline of a file, the file must be opened in the editor.
I	This is the Editor Area of the workbench where all the files open in the editor can be viewed.

Figure 2-2 shows the Debug perspective of the Eclipse Workbench. This perspective includes views that help you to debug the program.

For more information on views, see the Eclipse online help.

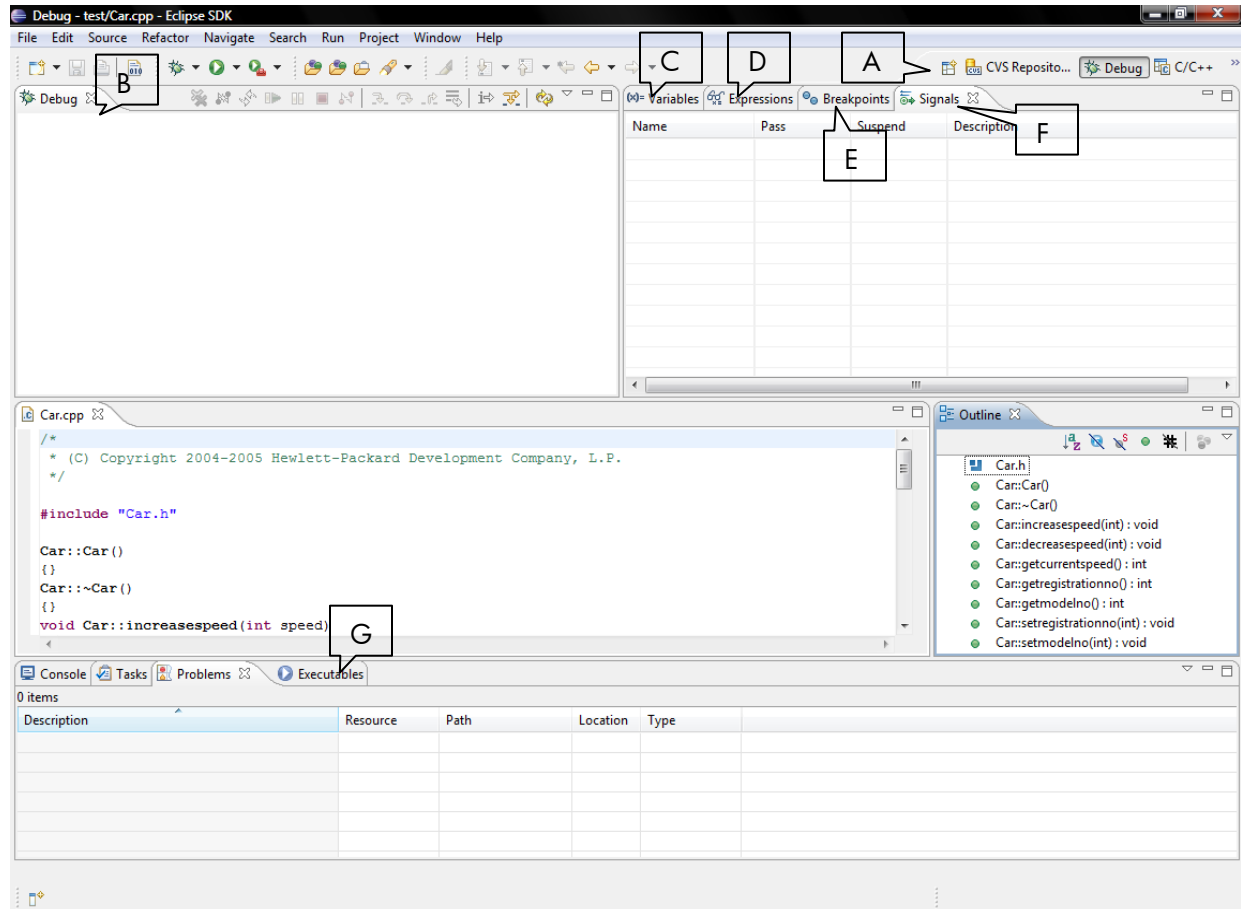


Fig 2-2: Debug Perspective

Table 2-2 provides a description of the Eclipse Workbench shown in Fig 2-2.

Table 2-2 Eclipse Workbench Description

Label	Description
A	This area of the workbench shows the selected perspective.
B	Debug view: displays debugging information in a tree hierarchy.
C	Variables view: displays information about the variables associated with the stack frame selected in the Debug view.
D	Expressions view: allows you to inspect data during debugging the program.
E	Breakpoints view: lists all the breakpoints currently set in the workspace.

Table 2 -2 Eclipse Workbench Description

Label	Description
F	Signals view: allows you to view the signals defined on the selected debug target and how the debugger handles the signals.
G	Executables view: lists all the executables present in the workspace.

Managing Tasks in SoftBench and HP-UX DEE

The following sections describe how tasks are performed in SoftBench and how they can be performed in HP-UX DEE:

- [Setting up a Project \(Page 9\)](#)
- [Modifying a Project Definition \(Page 12\)](#)
- [Creating Files within a Project \(Page 16\)](#)
- [Adding Existing Files to a Project \(Page 19\)](#)
- [Defining Targets \(Page 20\)](#)
- [Building Projects and Targets \(Page 20\)](#)
- [Interpreting Error Messages \(Page 22\)](#)
- [Running the Build on a Remote System \(Page 23\)](#)
- [Running Tools \(Page 23\)](#)
- [Customizing your Environment \(Page 23\)](#)
- [Getting Help \(Page 25\)](#)
- [Repositories \(Page 25\)](#)

Setting up a Project

SoftBench enables you to do the following:

- [Creating a Project Using Project Build](#)
- [Creating a Project Using External Build](#)
- [Cloning a Project from an Existing Project](#)
- [Repartitioning a Project into two Projects](#)

Creating a Project Using Project Build

For 'Project Build' projects, SoftBench does the following:

- Manages the build instructions or the makefile for the project.
- Generates the makefile so that it can be used for external purposes.

In HP-UX DEE, you cannot create a project equivalent to the SoftBench 'Project Build' project, but you can migrate the project from SoftBench.

To migrate a 'Project Build' project from SoftBench to HP-UX DEE:

1. Do one of the following:
 - a) Use the 'Generate makefile' option in SoftBench and generate a makefile.
 - b) Write a makefile for the project after creating the project in HP-UX DEE.
2. Map a network drive for the remote system on your local system.
3. Create a Remote Standard Make C/C++ project in HP-UX DEE with the directory where the project is located on the remote system as the source directory.

NOTE: Remote Standard Make C/C++ project requires its makefile to be named as “makefile” only. Figure 2-3 shows the makefile icon.



Fig 2-3: makefile icon

Creating a Project Using External Build

To create ‘External Build’ projects, SoftBench requires you to write and maintain your makefiles or build scripts.

Creating ‘External Build’ projects in SoftBench is equivalent to creating ‘Remote Standard Make C/C++’ projects in HP-UX DEE.

To migrate an ‘External Build’ project from SoftBench to HP-UX DEE:

1. Map a network drive for the remote system on your local system.
2. Create a Remote Standard Make C/C++ project in HP-UX DEE with the directory where the project is located on the remote system as the source directory.

For more information about creating a Remote Standard Make project, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **HP C/C++ Remote Developer Plug-in User Guide** → **Tasks** → **Creating a remote project**.

NOTE:

1. To create a new Remote Standard Make C/C++ project in HP-UX DEE, create a new directory on the remote system for the project. To create a Remote Standard Make C/C++ project in HP-UX DEE with the existing source, use the existing directories containing the source or target files.
2. In HP-UX DEE, if you create a project on a directory that already has some files or directories in it, the files or directories automatically become a part of the project and are listed in the Project Explorer view of Eclipse. This feature of Eclipse bypasses the necessity to add files into the project from the directory on which the project is created. However, in SoftBench if you create a project on a directory that already has some files in it, these files do not become a part of the project. You must explicitly add the files to the project by selecting “Project: Add File(s) to Project...”.
3. Though the existing files or directories become a part of the HP-UX DEE project, it is the Makefile that decides which files will be considered for generating the executables for the project.
4. If you delete any files from your project, they will be permanently deleted from the remote system. To exclude files or directories in the project from being compiled, write your makefile such that the files that are not required will be excluded from being compiled.

Cloning a Project from an Existing Project

To clone projects, SoftBench uses an existing project definition as the base for a new project.

Cloning of projects is not supported in HP-UX DEE. However, you can create a new project with the same properties as an existing project. To view the properties of an existing project, select the ‘Properties’ option from the Context menu for the project.

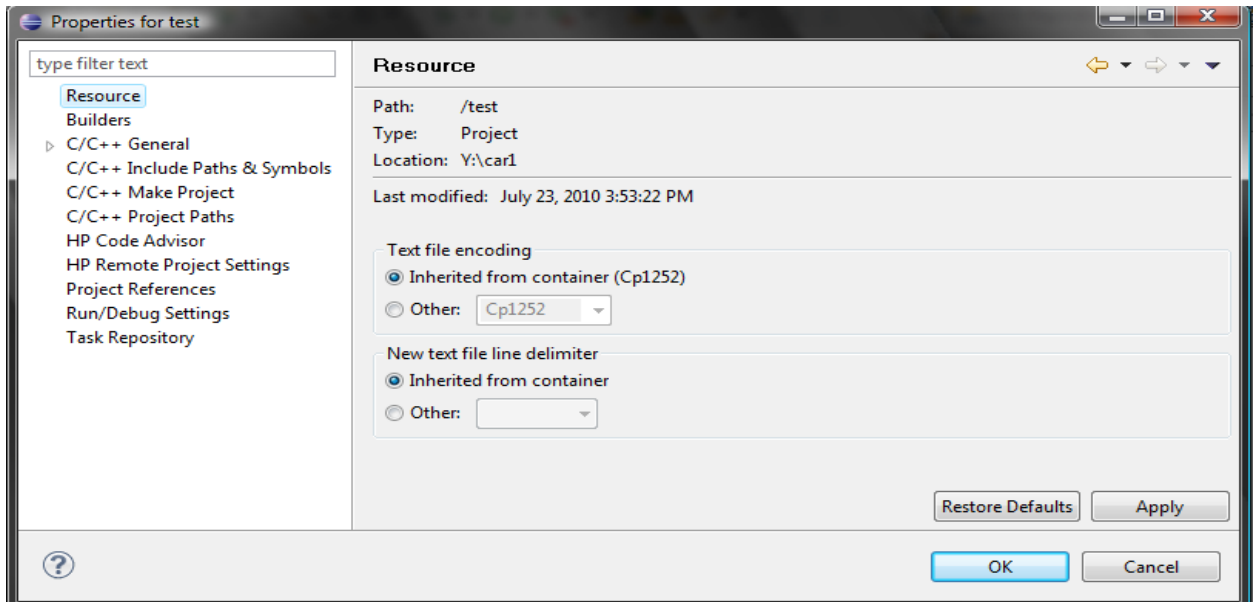


Fig 2-4: Properties Page

Repartitioning a project into two projects

SoftBench provides the ability to restructure a project, and create a second project from parts of the original project. During this process, you can choose whether the new project is a Subproject of the current project. Subprojects create build order dependency relationships between projects.

To restructure or repartition a project in HP-UX DEE, you can create a new project, move, or copy the required files from the old project into the new project.

To move a file from the old project to the new project:

1. Select the file to be moved.
2. Select **Move...** from the context menu.
The Move Resources dialog box appears, which provides a list of open projects in the workspace.
3. Select the project where you want to move the file and click **OK**.

To copy a file from the old project to the new project:

1. Select the file to be copied.
2. Select **Copy** from the context menu.
3. Right-click the project you want to copy the file to and select **Paste** from the context menu.

In HP-UX DEE, the files on the remote system are physically copied or moved to the new location.

You cannot create Subprojects in HP-UX DEE. However, you can create build order dependency relationships across projects in HP-UX DEE. SoftBench enables you to select the following:

- If you want to build the Subproject before building the project, you can create the required projects in HP-UX DEE and establish a build order dependency relationship across the projects.
- If you do not want to build the Subproject before the building the project, you need not establish the build order dependency relationship in HP-UX DEE.

To create build order dependency relationships across projects, create the project that needs to be built. When you create the second project, on the "Remote C/C++ Make Project Settings" wizard (Figure 2-5), click the **Projects** tab. A list of open projects in the workspace under the heading Referenced C/C++ Projects appears. Check the checkbox for the project you want to build first. After creating the second project, when you invoke build on the second project, the referenced project will be built first followed by the second project.

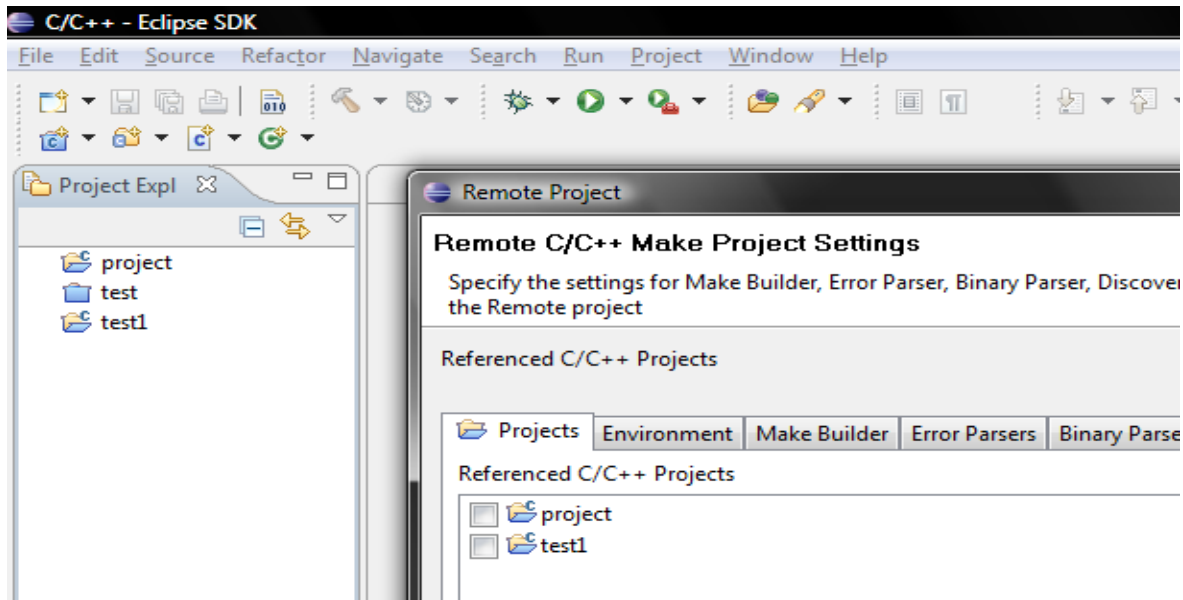


Fig 2-5: Referencing during Project creation

If you have already created the project that needs to refer to another project, complete the following steps:

1. Go to the Properties Page for the project.
2. Select **Project References**.
A list of open projects in the workspace, which the current project can refer to, appears on the right pane of the Properties page.
3. Check the check box for the required project and click **OK**.

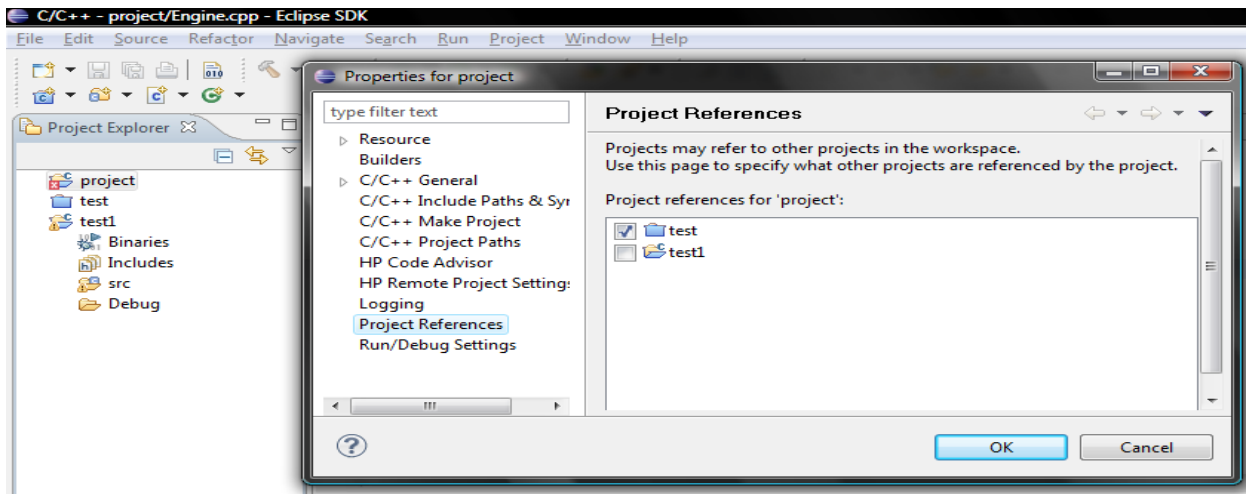


Fig 2-6: Referencing using Properties Page

Modifying a Project Definition

SoftBench enables you to modify the following attributes:

- Project name
- local workspace root
- Project type
- Location of Static database.

HP-UX DEE provides the **Rename...** option from the Context menu for the project to rename your project. You can modify only the name of the project but not its type because HP-UX DEE allows you to create only Remote Standard Make C/C++ projects.

NOTE:

1. The type of project created cannot be changed in HP-UX DEE.
2. The location of the project in HP-UX DEE cannot be changed. This property cannot be modified in the Properties Page of Eclipse (Figure 2-7).

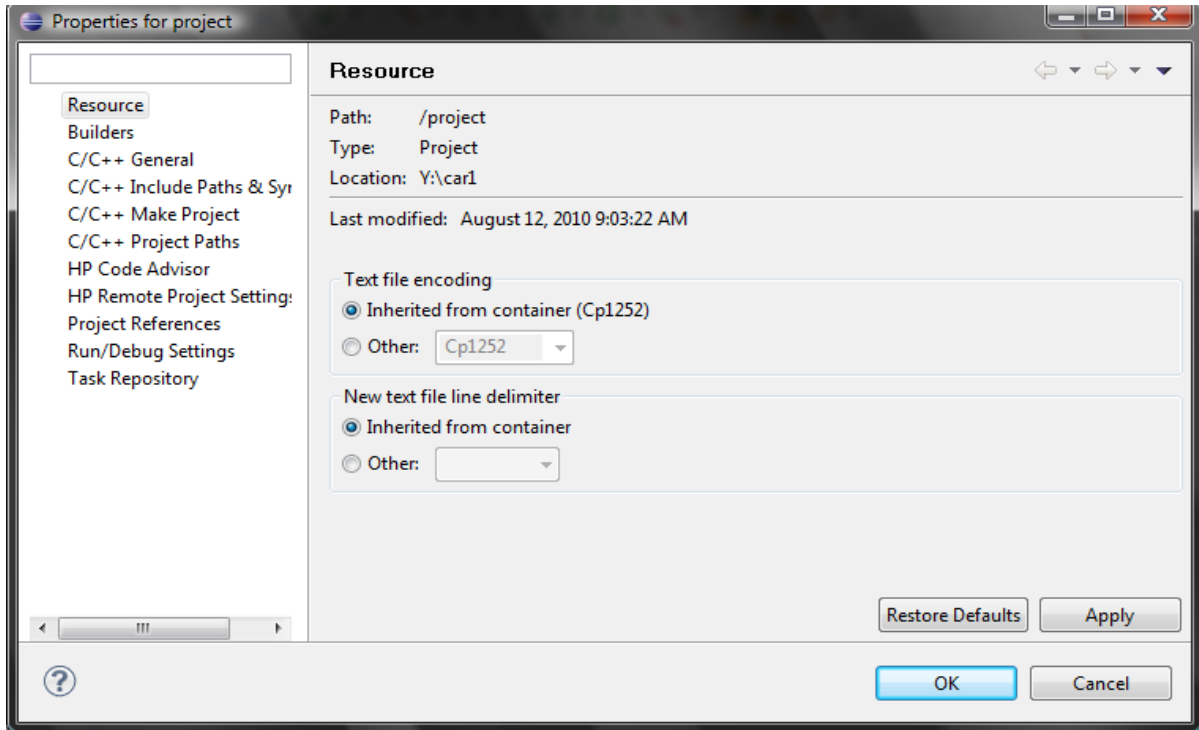


Fig 2-7: Properties Page of a HP-UX DEE Project

To change the properties of the project in HP-UX DEE, select **Properties** in the context menu for the project.

The following is a list of properties in an HP-UX DEE project that can be modified:

- Code Style: To change the code style, select the **Enable project specific settings** checkbox (see Figure 2-8).

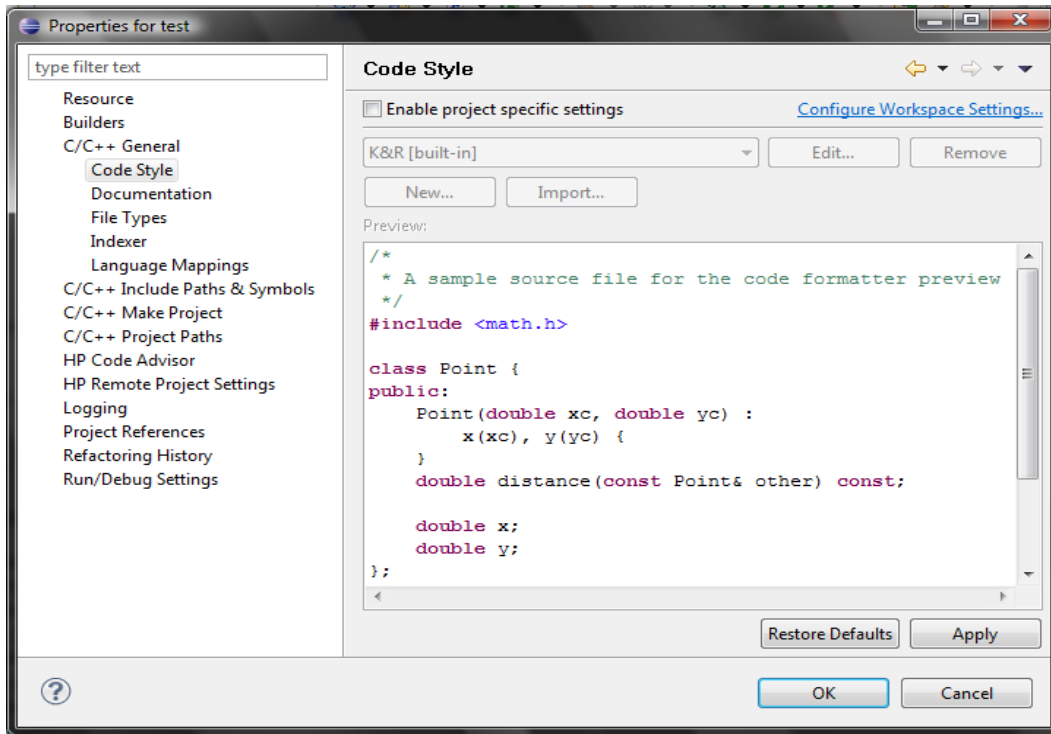


Fig 2-8: Code Style Properties Page

- Indexer: The C/C++ indexer uses the parser to create a database of your source and header files that provides the basis for tasks, such as C/C++ search, navigation features, and parts of content assist. The Indexer Property page enables you to select the required indexer.
- C/C++ Include Paths and Symbols: This page displays the directories that include the basic, required header files. The directories shown in Figure 2-9 are included in all the HP-UX DEE projects by default. You can add other directories, edit, or remove the existing directories.

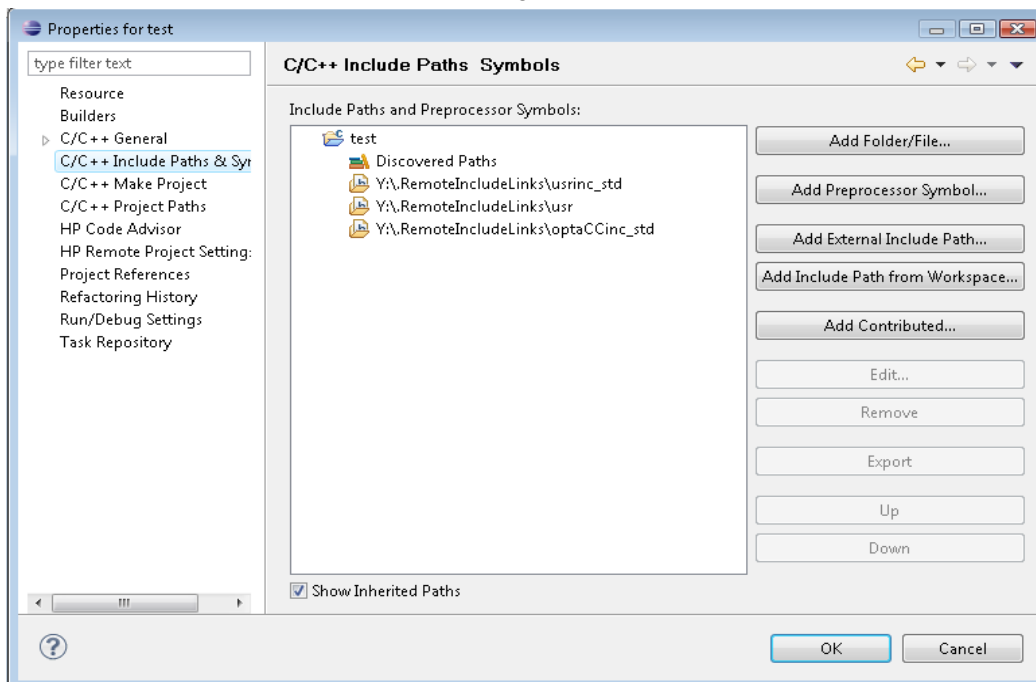


Fig 2-9: C/C++ Include Paths & Symbols Properties Page

- C/C++ Make Project: This page provides options related to the Build Settings. You can select **Build Command**, which by default is 'make'. To stop the build process when it encounters the first error, select the Stop on first build error check box. You can also select the **Make build target** for build and clean.

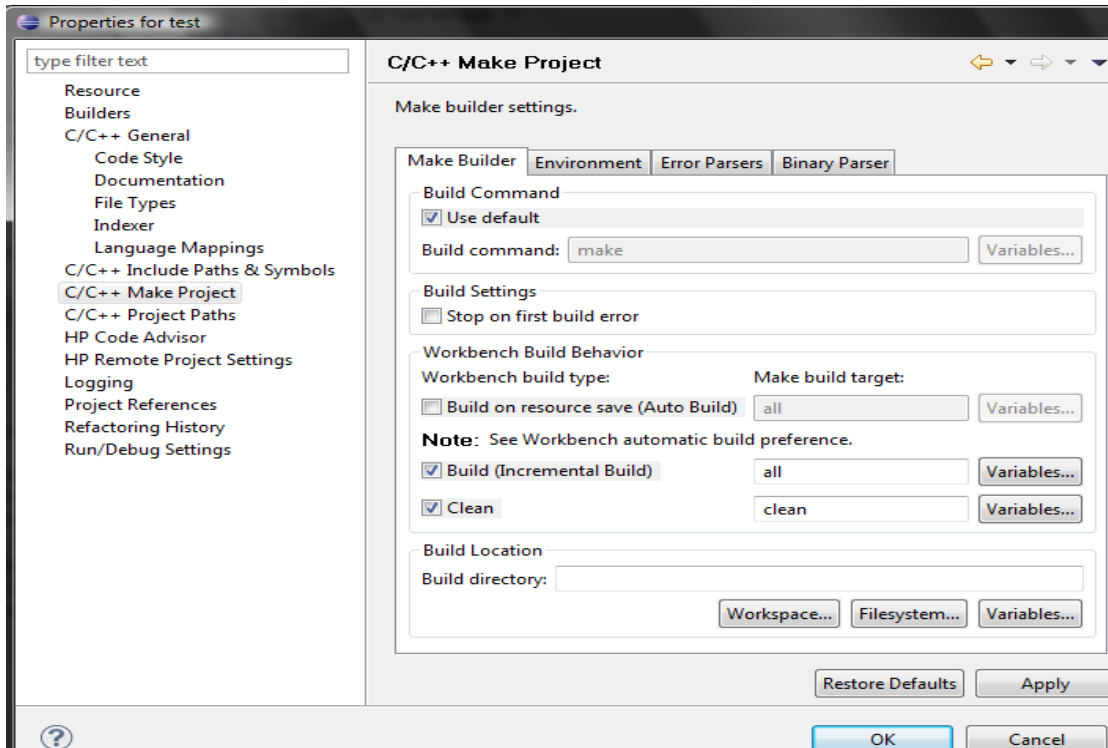


Fig 2-10: C/C++ Make Project Properties Page

- HP Code Advisor (cadvice): The text boxes in this page are populated with the default values. You can modify them as appropriate.

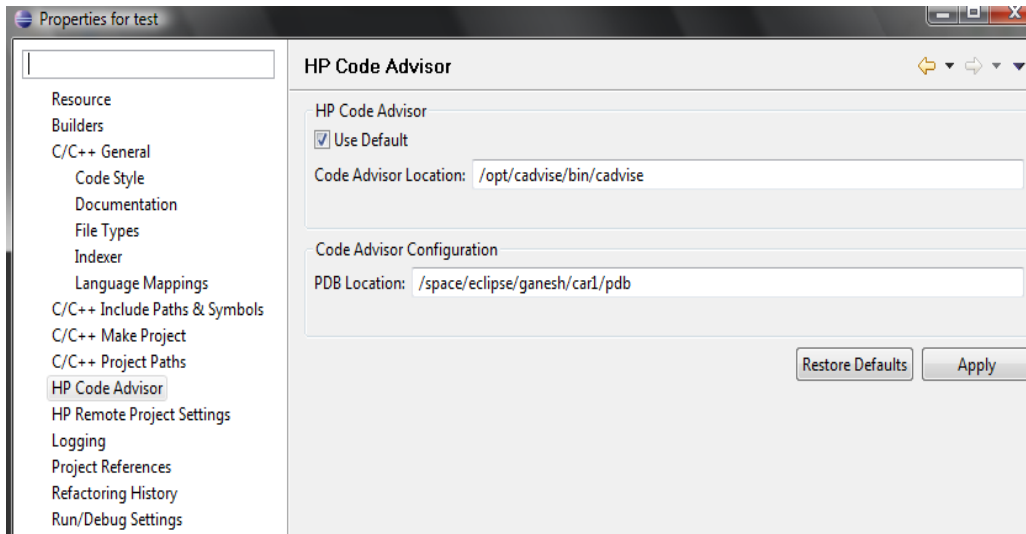


Fig 2-1 1: HP Code Advisor Properties Page

- HP Remote Project Settings: You can modify the Username and Display IP Address, which is the IP address of the local system.

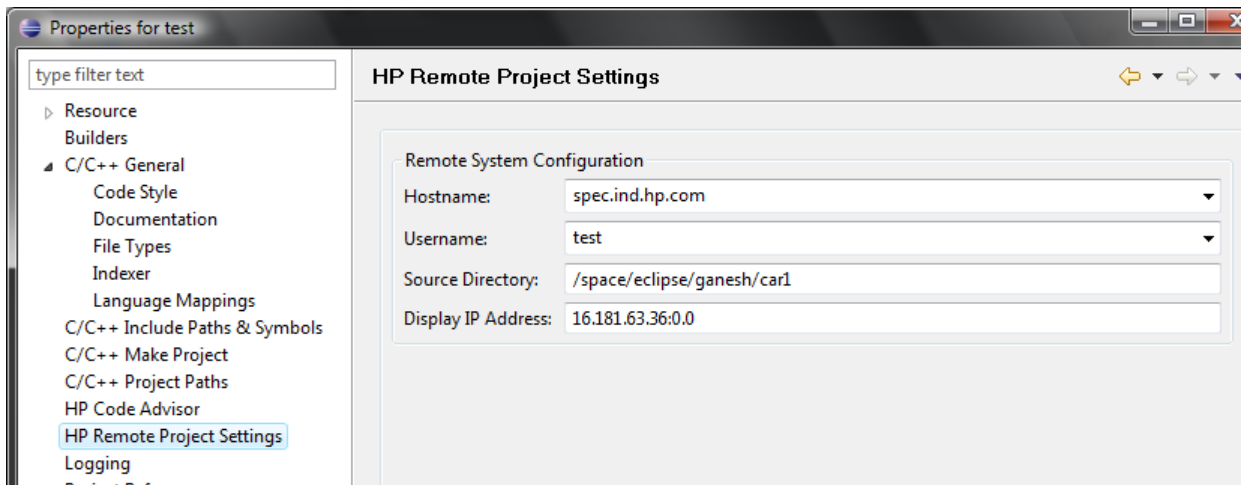


Fig 2-12: HP Remote Project Settings Properties Page

- Logging: Provides the location of the build log file.
- Project References: Provides a list of projects open in the workspace that can be referenced for the current project.

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **C/C++ Properties** → **C/C++ Project Properties**.

NOTE: You can access the help information on any particular Properties page by clicking the help icon after navigating to the required page.



Fig 2-13: Help icon

Creating Files within a Project

In Eclipse, you can invoke the File Creation wizard in one of the following ways:

- Click **File** → **New**. Select the type of file you want to create (source file or header file) as shown in Figure 2-14.



Fig 2-14: Using File menu to invoke File Creation Wizard

- Click **New** in the toolbar and select the type of file you want to create.

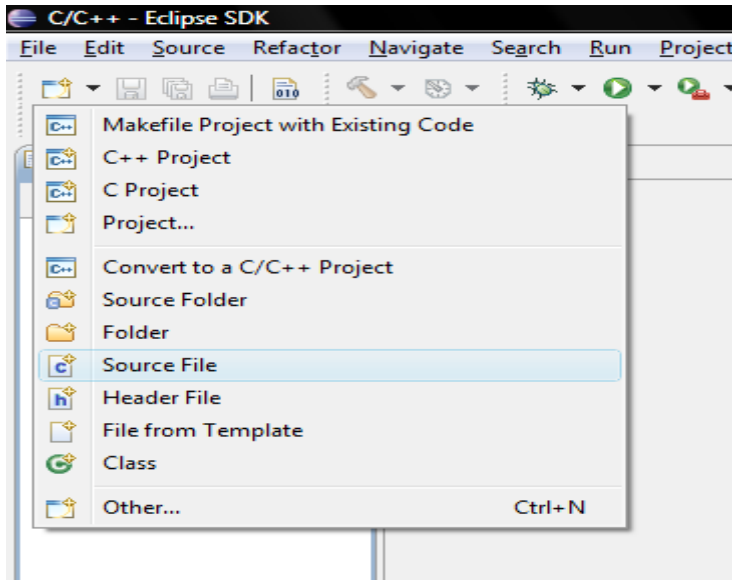


Fig 2-15: Using New icon to invoke File Creation Wizard

- From the context menu for the project, select **New** and select the type of file you want to create.

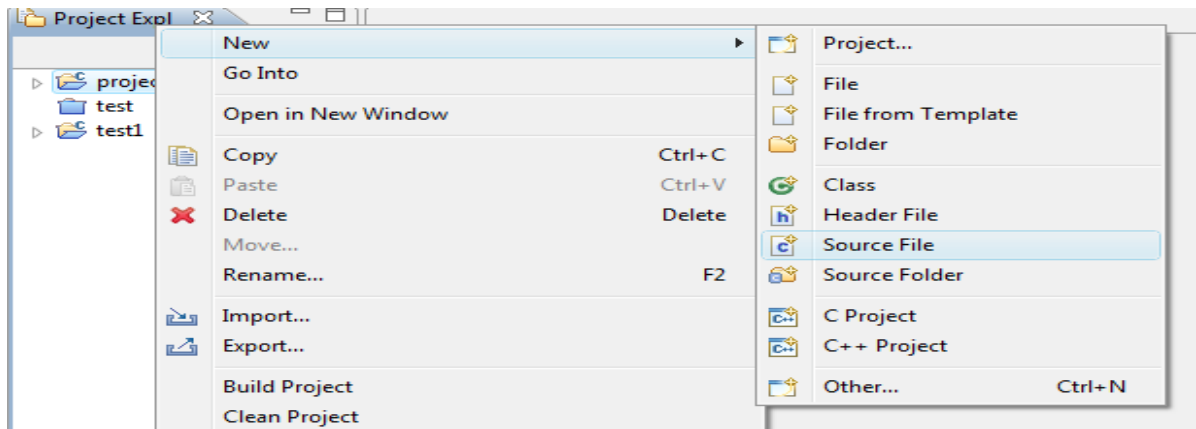


Fig 2-16: Using Context menu to invoke the File Creation Wizard

Source File

Selecting the Source file option opens the New Source File wizard.

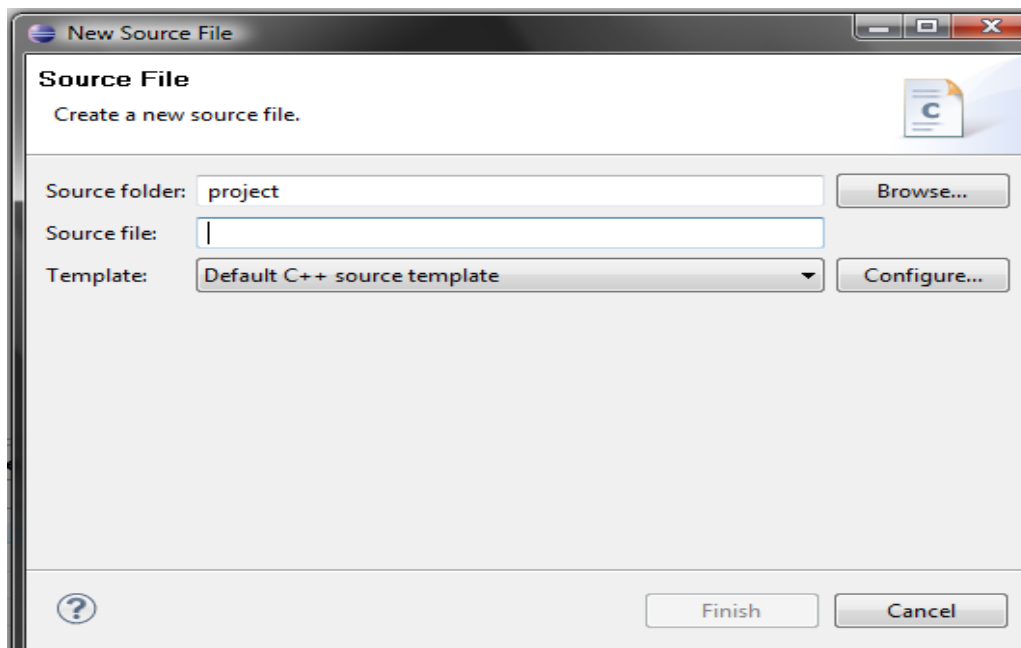


Fig 2-17: Source File Creation Wizard

- Source folder: Enter the location where you want to create the new source file.
- Browse: Select the location of the new source file.
- Source file: Enter the name of the new file. Ensure that the name of the file has the appropriate extension.
- Template: Select one of the options from the drop-down list– None, Default C++ source template, or Default C source template.
- Configure: Click this button to configure the templates as required.
- Finish: This button is enabled only after you enter the correct data in the wizard. Click **Finish** to create the new file and open it in the default editor.

Header File

Selecting the Header file option opens the New Header File wizard.

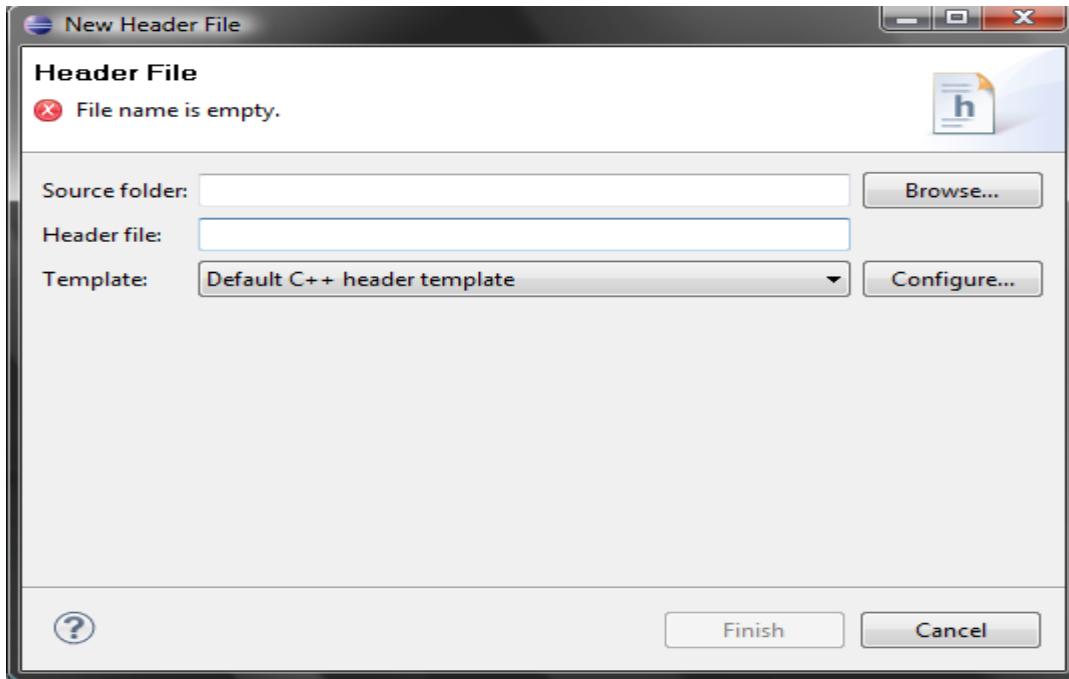


Fig 2-18: Header File Creation Wizard

- Source folder: Enter the location where you want to create the new header file.
- Browse: Select the location of the new header file.
- Header file: Enter the name of the new file. Ensure that the name of the file has the appropriate extension.
- Template: Select one of the options from the drop-down list– None, Default C++ header template, or Default C header template.
- Configure: Click this tab to configure the templates as required.
- Finish: This tab is enabled only after you enter the correct data in the wizard. Click **Finish** to create the new file and open it in the default editor.

NOTE:

1. If you select a project before invoking the file creation wizard, the Source folder text box in the wizard will be populated with the name of the project or folder. Otherwise, the Source folder textbox will be empty.
2. If you select the C/C++ source/header template but provide a filename with no extension, the created file will be a text file.

Adding Existing Files to a Project

The SoftBench vi editor and SoftBench XEmacs have the menu commands **File: Save** and **File: Save As...**, which enable you to add an existing file to the project listed in the editor's title bar. To add files individually or to add files in groups, select **Project: Add File(s) to Project...** from the main SoftBench window.

To add files from the local drive or mapped network drive in your system to your project, use the 'Import' feature in Eclipse. For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **Workbench User Guide** → **Getting Started** → **Basic tutorial** → **Importing files** → **Import Wizard**.

Defining Targets

For External build projects in SoftBench, you must create the Target and save the command to initiate the build.

In HP-UX DEE, you need to create a makefile, which will contain all the instructions required to build the project. In Eclipse, the Make Target view lists the targets you create for your projects (see Figure 2-19(a)). You can create Make targets for your project from this view by selecting **New** from the context menu for the project in this view (see Figure 2-19(b)).

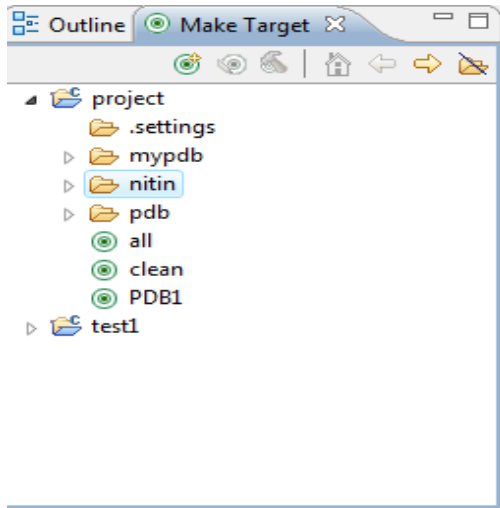


Fig 2-19(a): Make Target View

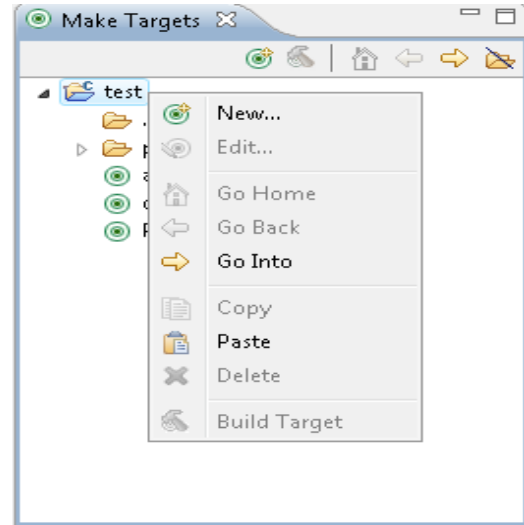


Fig 2-19(b): Context menu in Make Target View

To initiate building of your projects, double-click the targets. For example, if your makefile has a command 'all', which has the instructions to generate the binary from the source files, you can create a target for 'all'. On double-clicking this target, you can create the binary for the project.

The Project Explorer view in Eclipse also lists the intermediate files and the binaries of a project after it is built. For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **C/C++ Development User Guide** → **Tasks** → **Building projects** → **Creating a make target**. You can also click the **Create a Make Target** link under **Related Reference** at the bottom of the page.

Building Projects and Targets

In HP-UX DEE, you can build the project by creating make targets and double clicking on them to invoke the build.

You can also build the project using the following options:

- Build Project option from the context menu for the project
- Build Project option from the Project menu

A progress bar with **Cancel** appears. To cancel the build process, click **Cancel**.

NOTE: After a project references another project, select **Build Project** or **Clean Project** from the context menu or the project menu to build the project and its referenced project.

Setting the Compile Mode

In SoftBench, the Compile Mode is a prerequisite to perform debugging and static analysis. The compile mode decides which compiler flags will be used during the build process. The compile modes are:

1. Debug: instructs the compiler to prepare an object file for debugging.
2. Static: informs the compiler to prepare the object file for static analysis.
3. Optimized: informs the compiler to optimize the executable.

To modify the compiler flags in external build projects, select **Builder: Use External Build Command...** The External Build Command dialog box appears. Add the compiler flags in the Options text box.

The External Build Command dialog box is used to customize the build for external build projects.

HP-UX DEE does not have Compile Mode options. However, you can provide the compiler flags in the makefile, which contains the build instructions to be sent to the compiler.

You can also set environment variables to define the compiler flags. To set the environment variables, complete the following steps:

1. Go to the Properties page for the project.
2. Select **C/C++ Make Project → Environment**.
3. Define the variables.

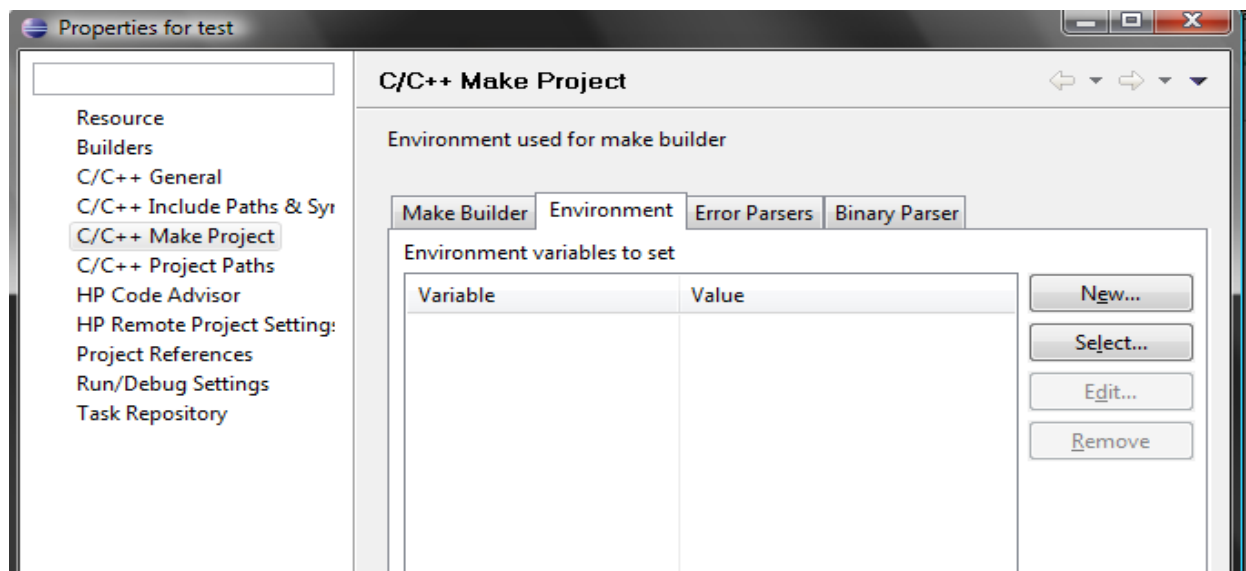


Fig 2-20: Environment tab in C/C++ Make Project Properties Page

Previewing the Build

SoftBench allows you to preview the build by simulating the build without performing the actions.

In HP-UX DEE, you can view the build output in the Console view only after building the project.

Compiling Instead of Building

Use the Build command to compile the source files into object files, which are then linked into a target. Use the Compile command to compile the source file into the object file.

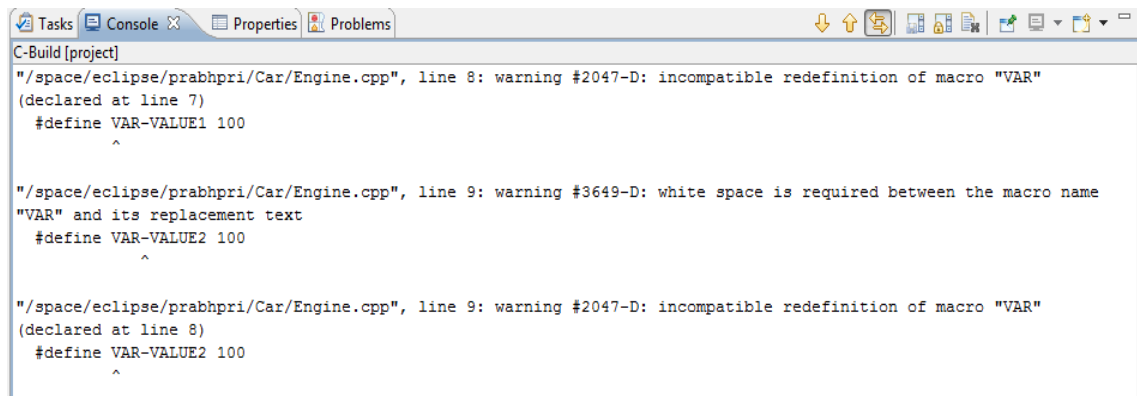
HP-UX DEE does not have the Compile command. During the build process, files are compiled into object files, which are linked to obtain the executable. The object files are displayed in the Project Explorer view of Eclipse.

To compile individual files, provide the command in the makefile. Create a Make Target to compile each file.

Interpreting Error Messages

In SoftBench, the error messages detected during build are displayed in the output browser. The error messages displayed in the output browser depend on the compiler you use. The error messages include the filename and line number to indicate where the error was detected. It also indicates the problem the compiler found.

In Eclipse, the error and warning messages detected during the build are displayed in the Console view (see Figure 2-21) and in the Problems view (see Figure 2-22). Double-clicking the error or warning message in the Problems view enables you to reach the point in the source code where the error or warning occurred.



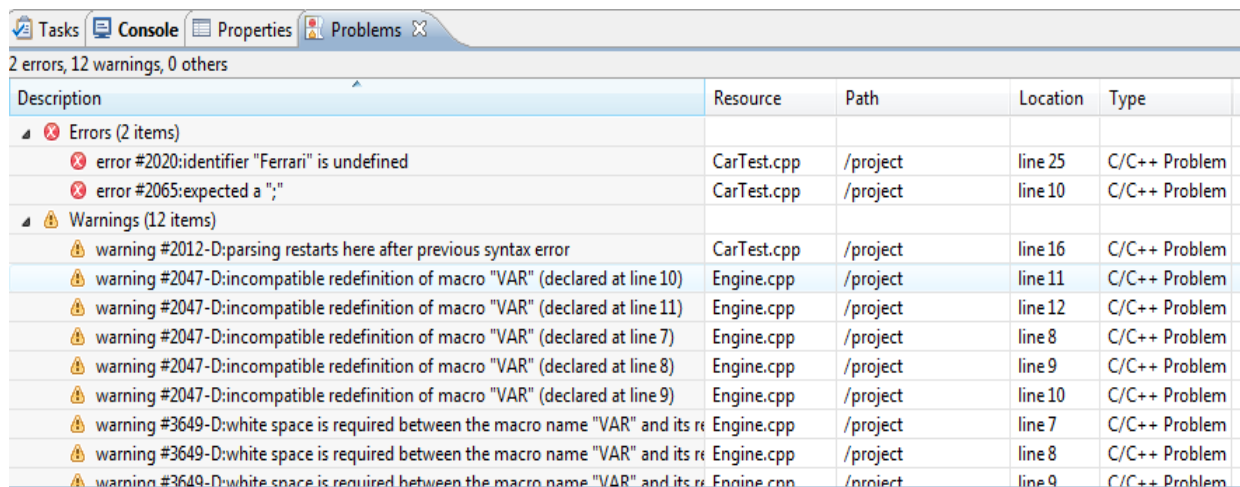
```
C-Build [project]
"/space/eclipse/prabhpri/Car/Engine.cpp", line 8: warning #2047-D: incompatible redefinition of macro "VAR"
(declared at line 7)
  #define VAR-VALUE1 100
      ^

"/space/eclipse/prabhpri/Car/Engine.cpp", line 9: warning #3649-D: white space is required between the macro name
"VAR" and its replacement text
  #define VAR-VALUE2 100
      ^

"/space/eclipse/prabhpri/Car/Engine.cpp", line 9: warning #2047-D: incompatible redefinition of macro "VAR"
(declared at line 8)
  #define VAR-VALUE2 100
      ^
```

Fig 2-21: Console View

Figure 2-22 shows the Problems View in HP-UX DEE.



Description	Resource	Path	Location	Type
2 errors, 12 warnings, 0 others				
Errors (2 items)				
error #2020:identifier "Ferrari" is undefined	CarTest.cpp	/project	line 25	C/C++ Problem
error #2065:expected a ","	CarTest.cpp	/project	line 10	C/C++ Problem
Warnings (12 items)				
warning #2012-D:parsing restarts here after previous syntax error	CarTest.cpp	/project	line 16	C/C++ Problem
warning #2047-D:incompatible redefinition of macro "VAR" (declared at line 10)	Engine.cpp	/project	line 11	C/C++ Problem
warning #2047-D:incompatible redefinition of macro "VAR" (declared at line 11)	Engine.cpp	/project	line 12	C/C++ Problem
warning #2047-D:incompatible redefinition of macro "VAR" (declared at line 7)	Engine.cpp	/project	line 8	C/C++ Problem
warning #2047-D:incompatible redefinition of macro "VAR" (declared at line 8)	Engine.cpp	/project	line 9	C/C++ Problem
warning #2047-D:incompatible redefinition of macro "VAR" (declared at line 9)	Engine.cpp	/project	line 10	C/C++ Problem
warning #3649-D:white space is required between the macro name "VAR" and its replacement text	Engine.cpp	/project	line 7	C/C++ Problem
warning #3649-D:white space is required between the macro name "VAR" and its replacement text	Engine.cpp	/project	line 8	C/C++ Problem
warning #3649-D:white space is required between the macro name "VAR" and its replacement text	Engine.cpp	/project	line 9	C/C++ Problem

Fig 2-22: Problems View

Table 2-3 lists the columns in the Problems View.

Table 2 - 3 Problems View Description

Problems View Column heading	Description
Description	Describes the error or warning.
Resource	Indicates the name of the file where the error or warning occurred.
Path	Indicates the path where the resource can be found in the local workspace. This path is internally linked to the appropriate path on the remote system.
Location	Provides the line number in the source file where the error or warning occurred.
Type	Indicates the type of error.

NOTE: The errors and warnings shown in the Console view will be parsed and displayed in the Problems view.

When errors and warnings are detected after the build, error and warning indicators appear on the icons of the appropriate files. For example, Figure 2-23 shows a red mark that appears for files containing errors and a yellow triangle for files that contain warnings.

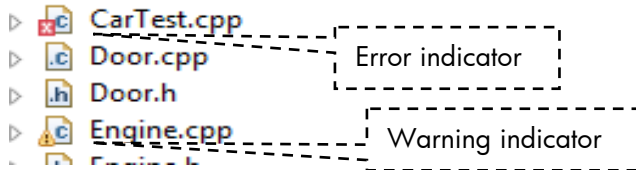


Fig 2-23: Error and Warning Indicators

When the file containing an error or warning indicator is opened in the editor, similar error or warning indicators appear in the Annotation margin of the editor. When you place the mouse pointer on the error or warning indicator, the error or warning message is displayed.

Running the Build on a Remote System

SoftBench enables you to build your project on a remote computer.

In HP C/C++ Remote Developer Plug-in for Eclipse, source files are compiled in the remote system where they are located. As a result, the build takes place on the remote system.

Running Tools

To access tools in SoftBench, you can use a menu command or the toolbar that includes icons for the tools available in SoftBench. You can start a tool by selecting the resources and invoking the tool from the toolbar or from the menu command. Alternatively, you can start the tool and then select the resources from the tool's menu commands.

Eclipse provides menu commands and a toolbar containing icons for the tools available in Eclipse. In Eclipse, you must select the resources before starting the tools. For example, to start the editor, you need to select the file you want to open using the editor. Similarly, you need to select the executable or the project before you start the debugger.

Customizing your Environment

SoftBench includes the **Views** and **Options** menus, which allow you to customize the SoftBench Environment. You can show or hide the toolbar, the target graph, change the tool icons displayed in the toolbar, set tool preferences,

and so on. SoftBench also provides you with a Custom menu, which enables you to create your own commands on the SoftBench toolface.

Eclipse allows you to customize the perspective you are using. You can select the menu commands and toolbar icons to be displayed. To customize the perspective view, select **Customize Perspective...** from the Window menu. The Customize Perspective - <perspective name> window appears, where you can customize the perspective.

Adding and Removing Tool Icons

In SoftBench, you can use **Toolbar Setup...** from the Options menu to add and remove tool icons in the toolbar.

To add or remove tool icons in Eclipse, you need to customize the perspective. In the Customize Perspective window, click the Tool Bar Visibility tab.

Choosing Tool Preferences

Some of the SoftBench tools allow you to select more than one tool. For example, SoftBench supports two editors, the SoftBench XEmacs Editor and the SoftBench vi Editor. To set the tool preferences, SoftBench provides the **Tool Preferences...** under the Options menu.

In Eclipse, if the tool provides more than one choice, you can select the default choice from the available choices. For example, to select the type of editor, select **Open With** from the context menu for the file. From the Open With submenu that pops, select the editor to open the file. To modify file associations, select **General → Editors preference page** (to open the Preferences page, go to **Window → Preferences**).

Accessing Distributed Data and Tools

You can install SoftBench in a distributed system, where the SoftBench is installed in one system but it is accessing data on another system in the network.

You can install Eclipse in your local workstation. Using HP-UX DEE, you can access resources on a remote HP-UX system through the network drive, create projects, and build them.

Exiting the Workspace

To exit HP-UX DEE, close Eclipse by clicking **Close** on the top right corner of the screen. The Confirm Exit dialog box appears. Click **OK** to confirm.

Restoring your Previous Session

When you close Eclipse, the current settings of the Eclipse workspace are saved. On restarting Eclipse, it restores the previous session automatically. Any editors that are open while closing Eclipse are opened on restarting Eclipse.

Getting Help

SoftBench provides an online help that can be accessed from the Help menu, which is provided in the main SoftBench window and in all the SoftBench tools. You can access the generic help file on any SoftBench tool, task-oriented help, or specific help on a selected window item. SoftBench allows you to view information about screen areas such as menu commands, input boxes, or dialog boxes, by moving the mouse pointer over the item, and then pressing the Help key (F1).

To access the online help in Eclipse, select **Help** -> **Help Contents**. The Eclipse online help page appears. The Contents column on the left pane enables you to navigate to a help topic.

In Eclipse, the Help key is F2. Eclipse provides a Help icon, shown in Fig 2-13, (a question mark) in every wizard and dialog used in Eclipse.

Repositories

SoftBench Configuration Management (SoftCM) helps you to manage software development projects that include multiple versions and configurations. Using SoftCM, you can retrieve and build any version of an application in a consistent, repeatable manner.

HP-UX DEE does not support Configuration Management. To create a project in HP-UX DEE, you must have the source code of the project on the remote system. To do this, you need to migrate your archive data stored in SoftCM to a repository, which is configured such that when a check-out is performed, the source code from the repository is loaded on the remote system at the required location where the HP-UX DEE project can be created. The check-ins and check-outs to the repository are performed outside the HP-UX DEE environment.

3 Using Editors

SoftBench provides SoftBench XEmacs and SoftBench vi editor for text editing purposes. SoftBench includes the SoftBench program editor, which is used for backward compatibility. When the editor is started from SoftBench, it contains project information. Any file you save using **Save** or **Save As...** is added to the file set of the current project. To save a file outside the project, use **Save Out Of Project** and **Save Out Of Project As...**

Eclipse does not include XEmacs, vi, or the program editor. It provides a general text editor and Java editor. If the C/C++ Development toolkit (CDT) is installed in Eclipse, you can use the C/C++ editor.

HP-UX DEE uses the C/C++ editor for its source files. Both the C/C++ perspective and Debug perspective provide an editor area where an editor can be opened. To open an editor in the editor area, select a file. Double-clicking the file in the project explorer view opens the file in the editor area using the default editor. You can also use the Open option from the context menu for the selected file.

Eclipse editors do not understand vi commands or XEmacs rules but provide syntax highlighting, and indicate the presence of errors or warnings through the error or warning indicators. The Eclipse editor also provides the Content Assist feature, a tool that helps you to reduce the number of keystrokes required to create your code. To invoke the content assist dialog, press the CTRL+Space bar keys.

Figure 3-1 shows the Content Assist with a set procedures being prompted.

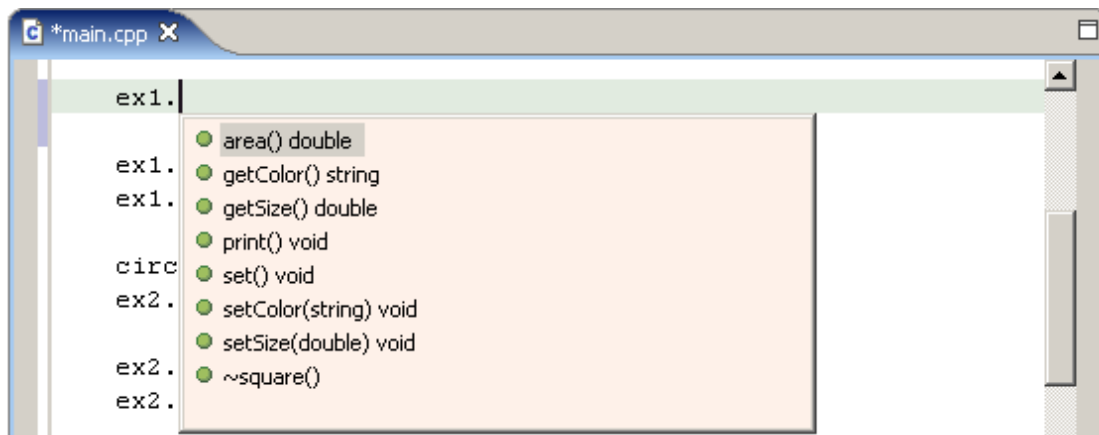


Fig 3-1: Content Assist

For more information, navigate to the following online Eclipse help page:

C/C++ Development User Guide → **Concepts** → **Code aids** → **Content assist**.

When you open a file in the default editor, the name of the file appears on the tab in the editor area. If the editor has unsaved changes, an asterisk appears in the tab to indicate that the file is not saved (see Figure 3-2(a)).

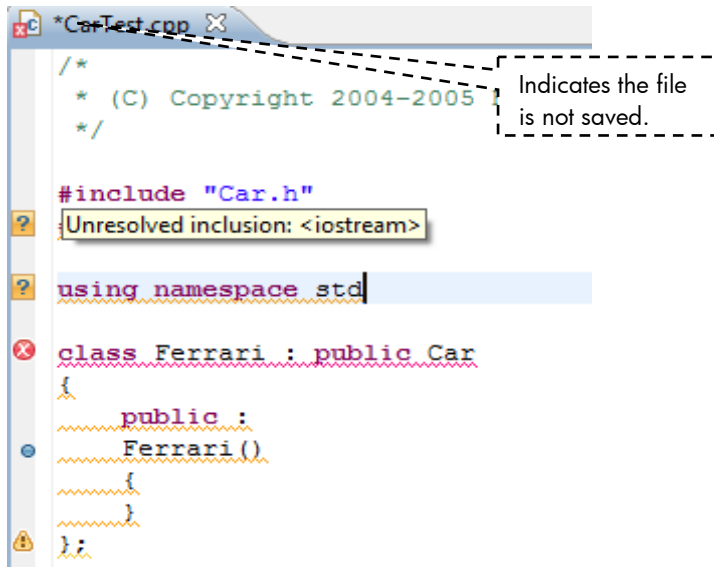


Fig 3-2(a): Editor Notations

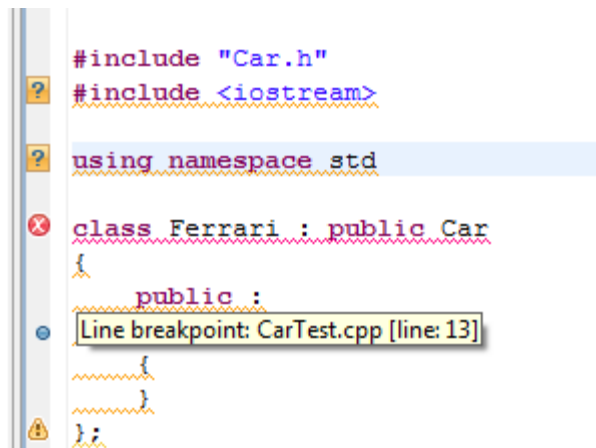


Fig 3-2(b): Editor Notations

The left pane in the editing space might contain icons that flag errors, warnings, or problems detected by the system. Icons also appear if you have created bookmarks, added breakpoints for debugging, or recorded notes in the Tasks view. You can view the details for any icon in the left margin of the editor (as shown in the first icon in Figure 3-2(a)) by moving the mouse cursor over it.

For more information about editors, navigate to the following pages in the online Eclipse help page:

In the Contents column, select:

1. **Workbench User Guide** → **Getting started** → **Basic tutorial** → **Editors and views** → **Editor**.
2. **Workbench User Guide** → **Getting started** → **Basic tutorial** → **Working with other editors**.
3. **Workbench User Guide** → **Concepts** → **Editors**.
4. **Workbench User Guide** → **Reference** → **User interface information** → **Views and editors** → **Editor area**.

The editor includes the following features:

- **Syntax highlighting:** The source code appears in different colors and fonts depending on the category of terms. For example, in Figure 3-2 (b) `include`, `using namespace`, are keywords in red colored font.
- **Content/code assist:** You can insert templates of your project to your source code instead of retyping the commonly used snippets of code.
- **Integrated debugging features:** The Debug Perspective also provides for the Editor area where the files being debugged are opened. You can add, remove breakpoints, edit the file during debugging and so on.

You can customize editor operations by modifying the Editor Preferences page. For more information on the Editor Preferences page, navigate to the following pages in the online Eclipse help page:

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **C/C++ Preferences** → **Editor preferences**.

In Eclipse, the **Save** option saves the changes to an existing file. This option is enabled only if the file in the editor has any unsaved changes. The **Save As...** option saves the changes to a file in any other open project in the workspace. The file is saved in the newly chosen project and the file without the changes is retained in the old project. The **Save As...** option is enabled even if the file does not have any unsaved changes.

Eclipse does not allow you to save files outside the projects.

To close a file opened in the editor, click the close button (X) on the selected editor tab.

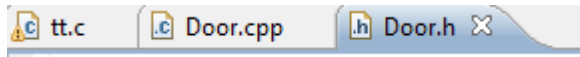


Fig 3-3: Editor tabs

To close a file selected in the editor area, select **Close** from the File menu. The **Close All** option in the File menu closes all the open editors. If you close a file with unsaved changes, you will be prompted to save the file before closing it.

Configuring the Editor

In SoftBench, you can select the default editor using **Options: Tool Preferences...**. Thereafter, every time you invoke the editor, it will open in the default editor.

In HP-UX DEE, the default editor for a file is selected depending on its extension. For example, if the file has .txt as its extension, the file opens in a text editor. If the extension is .c, the file opens in the C/C++ editor.

To open a file in an editor, do one of the following:

- Select **Open With** from the context menu of the file.
- Select the type of editor from the available list of editors. Double-click the file to open it with the selected editor.

Editing Multiple Files

Eclipse allows you to open multiple files in multiple editors in the editor area. However, you can edit only one file at a time. Eclipse also allows you to have multiple files in the editor area with unsaved changes. You can use the **Save All** option from the File menu to save all the files (see Figure 3-4).

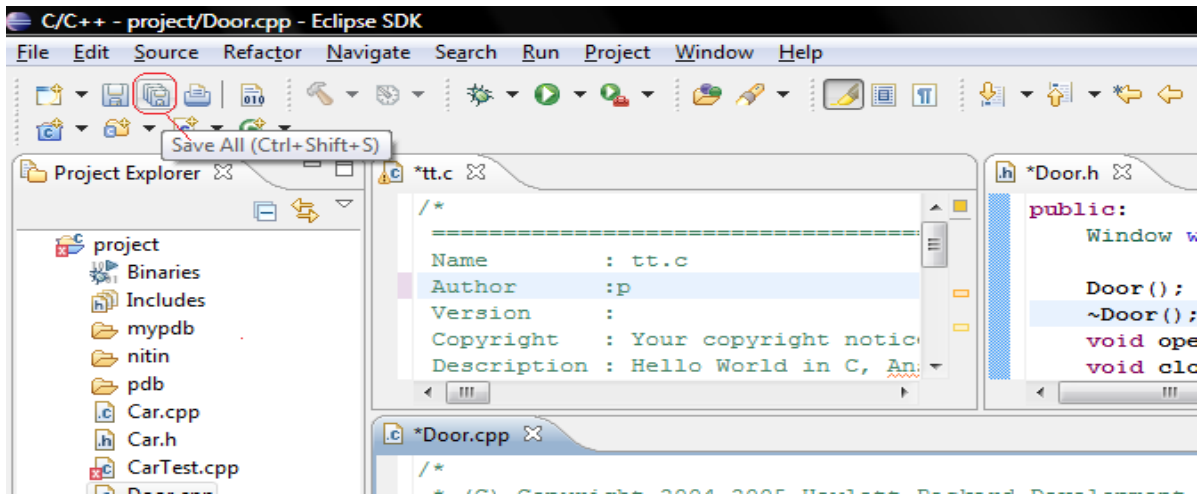


Fig 3-4: Save All icon

You can stack multiple editors in the editor area (see Figure 3-5). There are multiple tabs with filenames, where each tab represents the editor for one file.

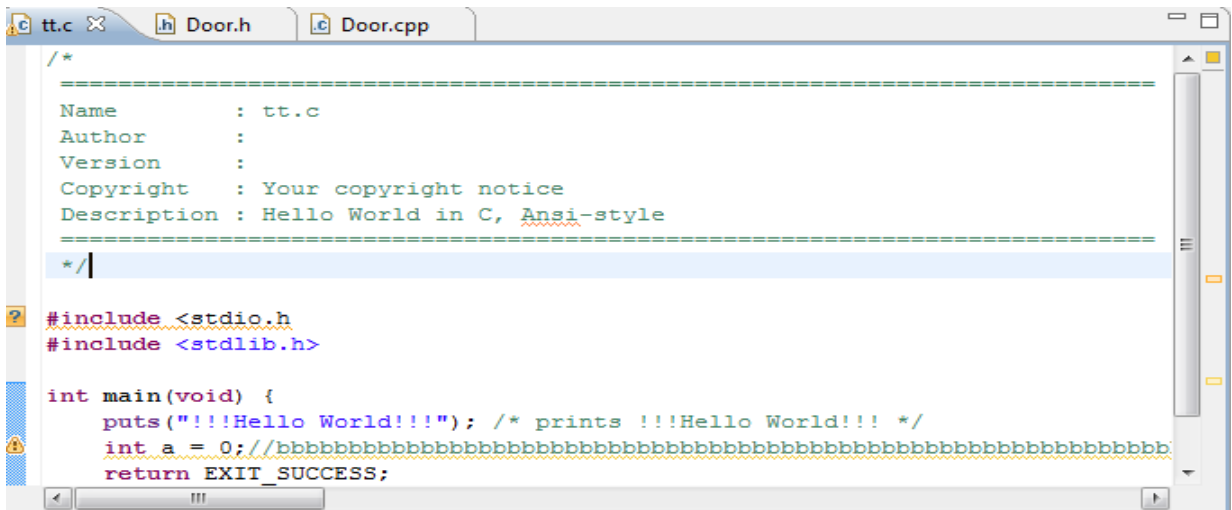


Fig 3-5: Multiple Open Files Stacked in Editor Area

You can also tile editor sessions within the editor area to view source files side by side. To tile editor sessions, select an editor tab, press the left mouse button and drag the selected editor over the left, right, top or bottom border of the editor area.

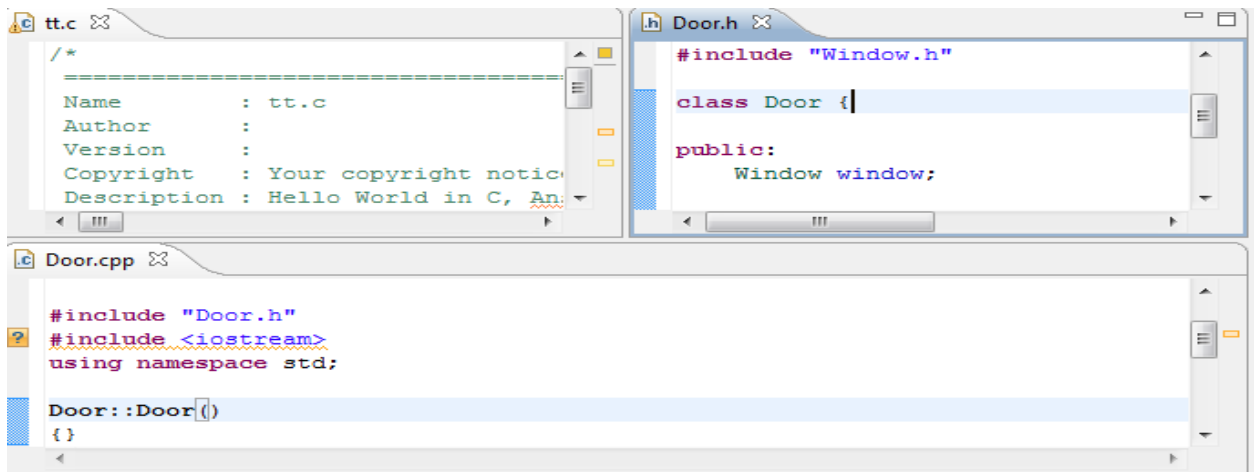


Fig 3-6: Multiple Open Files placed side by side in the Editor Area

To have the editor display one file at a time, modify the Editor Preferences page.

For more information on the Editor Preferences page, navigate to the following page in the online Eclipse help page:

In the Contents column, select **Workbench User Guide** → **Reference** → **Preferences** → **Editors**.

Managing Files and Projects Using the Editor

Both SoftBench and Eclipse enable you to invoke tools from the editor to perform functions. To perform these functions, select the options from the context menu in the editor.

Table 3-1 lists the tasks that can be performed from the editor.

Table 3-1: Editor-related Tasks

Task	SoftBench	Eclipse
Using Configuration management	Supported	Supported. This support is available if the repository used is CVS. Select Team in the context menu.
Compare Revision	Supported	Supported
Setting breakpoints	Supported	Supported
Cut-copy-paste actions	Supported	Supported
References	Supported	Supported
Declarations	Supported	Supported
Syntax highlighting	Supported	Supported
Find and Replace	Supported	Supported
Building the Project	Supported	Supported. To build the project, select Build Make Targets .
Compiling the file	Supported	Supported. To compile files, select Build Make Targets .
Compare file with local history	Not supported	Supported
Replace file with local history	Not supported	Supported
Generate code templates for getters and setters	Not supported	Supported
Create Make targets	Not supported	Supported
Run and Debug	Not supported	Supported

4 Using Code Advisor

SoftBench CodeAdvisor provides advanced code checking for C and C++ and can help you to locate errors that C and C++ compilers cannot detect. SoftBench CodeAdvisor does not duplicate the error-checking functions of the compilers. You can use SoftBench CodeAdvisor to identify issues such as potential heap corruption, dangling pointers, ambiguous initializations, and dependencies on system-specific compiler/linker behavior.

HP-UX DEE uses the HP Code Advisor (cadvice) to perform functions similar to the SoftBench CodeAdvisor. Cadvice is a static analysis tool for C and C++ programs, which reports programming errors in the source code. Cadvice takes C or C++ source files as inputs to identify errors, security violations, and so on, and stores the information in a program database (PDB). Cadvice is available for HP-UX developers.

Features of cadvice:

1. It is a powerful static code analysis tool that automatically diagnoses issues in a source program. Cadvice identifies defects during code development.
2. It detects a wide range of coding errors and potential problems such as memory leaks, used after free, double free, array/buffer out of bounds access, illegal pointer access, uninitialized variables, unused variables, format string checks, suspicious conversion and casts, out of range operations, C++ coding style warnings, and so on.
3. It detects security vulnerabilities in the source code such as buffer overflows, use of unsafe APIs, use of unsafe file path, unsafe data length argument, unsafe loop exit condition, unsafe use of tainted data, and so on.
4. It has a built-in coding guidelines checker, which diagnoses violations to the coding guidelines and emits appropriate diagnostic messages. You can modify the default set of rules or write your own set of rules to enforce coding guidelines.
5. It enables you to identify the potential problems in converting applications from 32-bit to 64-bit.

The result of the cadvice analysis can be stored in the PDB, which is a repository of errors, warnings, diagnostic messages, and metrics generated while analyzing the application.

For more information about using HP Code Advisor, see the *HP-UX DEE User Guide* at:

http://h21007.www2.hp.com/portal/download/files/unprot/hpc/eclipse-plugin/4_0_0/RDP%20User%20Guide.pdf

You can also navigate to the following page in the online Eclipse help page:

In the Contents column, select **HP C/C++ Remote Developer Plug-in User Guide** → **Getting started** → **HP Code Advisor Plug-in Tutorial**.

For more information on HP Code Advisor, see:

<http://www.hp.com/go/cadvice/>

5 Using Debugger

The SoftBench Debugger serves as a window interface to the HP debugger Distributed Debugging Environment (DDE), thereby enhancing and extending the DDE functionality.

HP-UX DEE uses the HP Remote GDB Debugger. In HP-UX DEE, you can start the debugger by selecting **Debug As** from the context menu, and then selecting **Debug Configurations** from the sub-menu. To debug the program, select **Debug Configurations** from the Run menu. Provide the appropriate information in the tabs in the Debug Configuration dialog box.

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **HP C/C++ Remote Developer Plug-in User Guide** → **Tasks** → **Running or debugging remote projects** → **Creating or editing a remote run or debug configuration**.

After configuring the Debug configurations for the executable, use the debug icon (marked in red in Figure 5-1) to initiate the debug process.

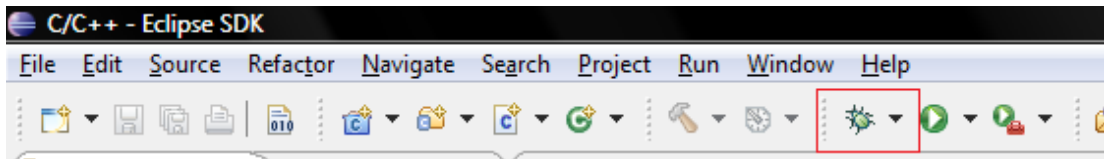


Fig 5-1: Debug Icon

In Figure 5-2, the debug configuration for the projects 'test' and 'ccc' have been defined. To debug the executables of the project 'test', click **test configuration**.

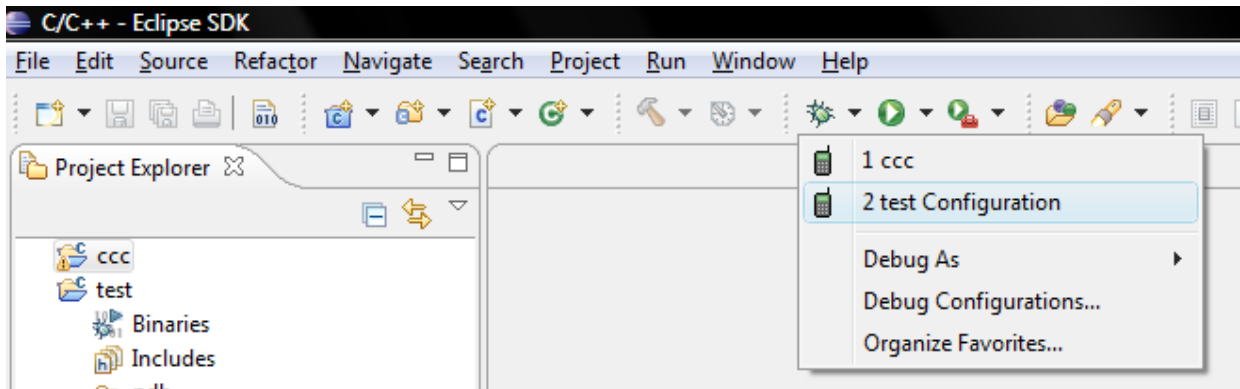


Fig 5-2: Debug Configurations

When the debugger is invoked, Eclipse changes to the Debug perspective. A tty terminal appears, displaying the output of the program being debugged.

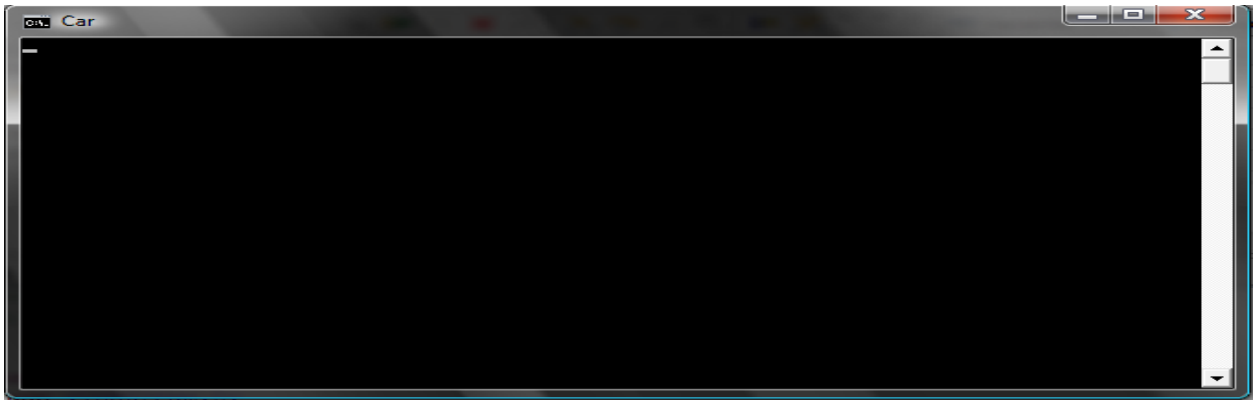


Fig 5-3: TTY terminal to display the output of the Debugged program

Stepping through a Program

The HP-UX DEE debugger pauses at the first executable statement in the main method irrespective of the presence or absence of breakpoints. You can modify this setting in the Debug Configurations dialog box.

After the debugger pauses, use the Debug view toolbar options to execute your program. For information about the Debug view toolbar options and Debug view context menu commands, navigate to the following page in the online help page for Eclipse.

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **CC++ views and Editor** → **Debug views** → **Debug view**.

The Debug view also displays the call stack and the thread stack.

The next statement to be executed by the debugger will have a 'Debug Current Instruction Pointer' pointing to it.

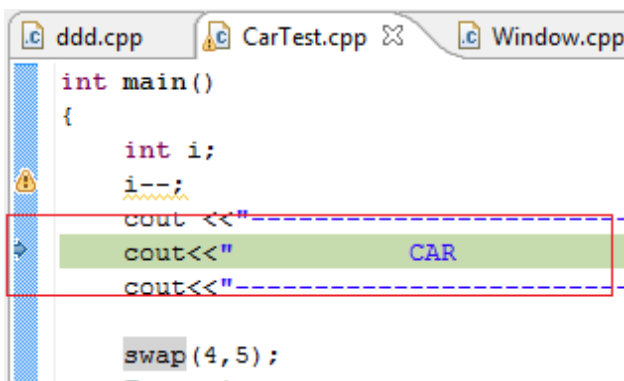


Fig 5-4: Debug Current Instruction Pointer

Interrupting a Running Program

To interrupt a program that is being debugged, terminate the program using the terminate option from the Debug view toolbar (see Figure 5-5).



Fig 5-5: Terminate Icon

Interacting with a Program

A tty terminal (see Figure 5-3) comes up when you start debugging a program in HP-UX DEE. You can interact with your program (provide the inputs it expects) through this terminal.

Examining and Modifying Data in your Program

To examine the value of a variable or modify the value of variables, use the Variables view in Eclipse.

To open the Variables view, go to **Window** → **Show view** → **Variables**.

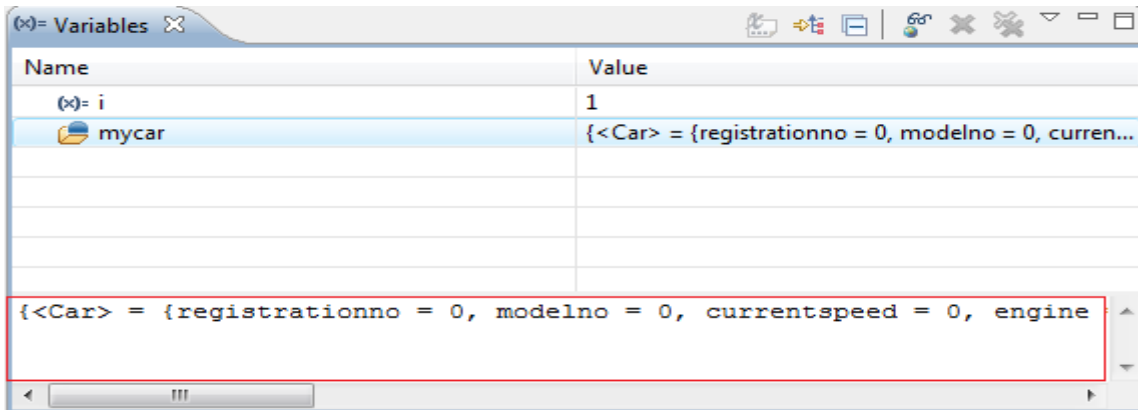


Fig 5-6: Variables View

Figure 5-6 shows the Variables view as seen in the Debug perspective.

The Variables View displays information about the variables associated with the stack frame selected in the Debug View. In Figure 5-6, the area marked with the red box is the Detail Pane, which displays detailed information about the selected variable.

The context menu for the selected variable and the context menu for the Detail Pane include commands that allow you to perform operations, such as modifying the value of the variable, viewing memory allocation, adding watchpoints, and so on.

For more information, see the Eclipse online help.

Eclipse also allows you to inspect expressions using the Expressions view. To open the Expressions view, go to **Window** → **Show view** → **Expressions**. The Figure 5-7 shows the Expressions view.

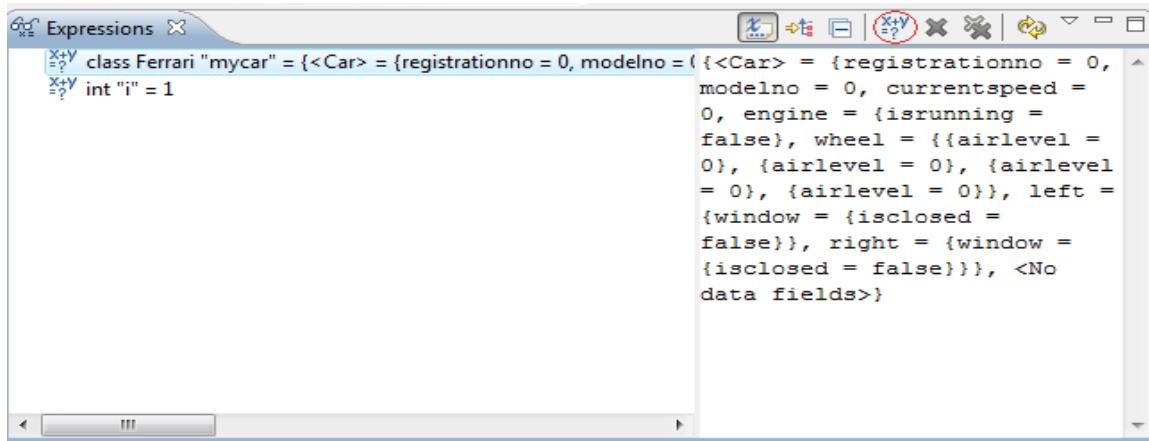


Fig 5-7: Expressions View

To add an item to the Expressions view, from the context menu in the view, do one of the following:

- Select **Add a Watch Expression**.
- Select the icon marked in red in Figure 5-7.

After you step through your program, you can monitor any changes to the expressions added in the Expressions view.

For more information, see the Eclipse online help.

Breakpoints

Breakpoints cause your program to stop executing at a specified location and return control to the debugger. In Eclipse, you can set breakpoints by right-clicking in the marker bar area on the left pane of the editor, beside the line where you want the program to be suspended, and then select **Toggle Breakpoint**. You can also double-click on the marker bar next to the source code line to set the breakpoint.

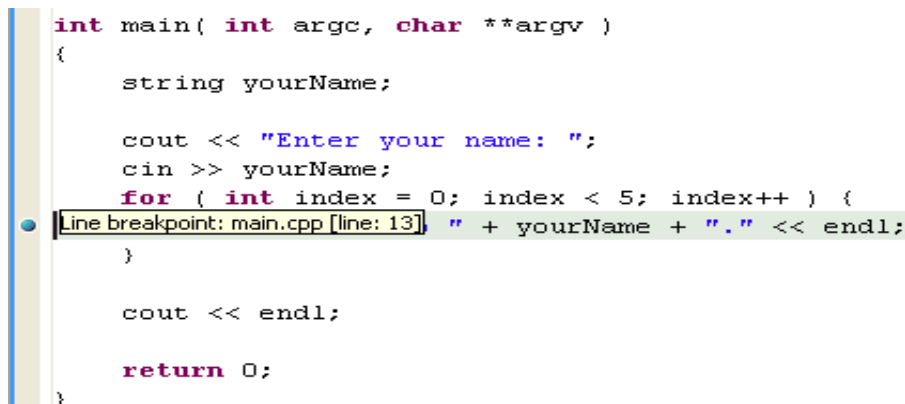


Fig 5-8: Breakpoint Indicator in Editor

For more information on breakpoints, navigate to the following page in the online help page for Eclipse.

In the Contents column, select **C/C++ Development User Guide** → **Concepts** → **Debug** → **BreakPoints**.

Eclipse also provides a BreakPoints view, where all the breakpoints in the workspace are listed.

To open the Breakpoints view, go to **Window** → **Show view** → **Breakpoints**.

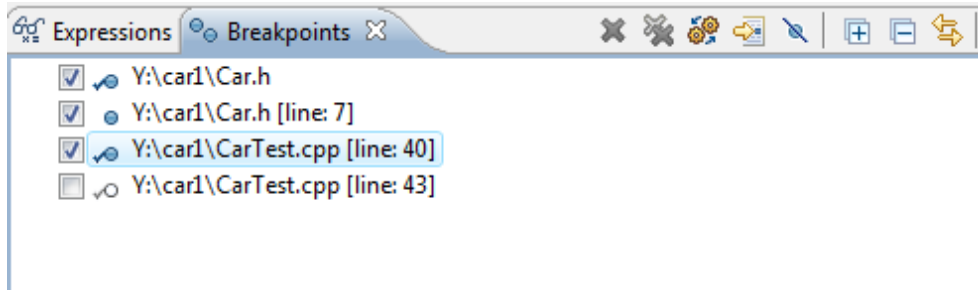


Fig 5-9: Breakpoints View

Double-clicking the breakpoint in the Breakpoints view takes you to the location of the breakpoint in the Editor. Clearing the check box for a breakpoint disables the breakpoint.

For more information, see the Eclipse online help.

To run from the current statement to be executed to the next breakpoint, click **Resume** in the Debug toolbar option.

WatchPoints

A watchpoint is a special breakpoint that stops the execution of an application whenever the value of a given expression changes. WatchPoints are associated with files.

For information about breakpoints, navigate to the following page in the online help page for Eclipse.

In the Contents column, select **C/C++ Development User Guide** → **Tasks** → **Running and debugging projects** → **Debugging** → **Using breakpoints, watchpoints and breakpoint actions**.

Tracing Program Flow

The tracing program flow feature is specific to the DDE debugger, and not supported in HP-UX DEE. As an alternative, you can step through your program, one executable statement at a time, to monitor the flow of the program.

Handling Signals and Events

Eclipse provides the Signals view to support signal handling. This is one of the views of the Debug Perspective. The Signals view enables you to view the signals defined on the selected debug target and how the debugger handles the signals.

For more information about using the Signals view, navigate to the following page in the online help page for Eclipse.

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **Debug views** → **Signals view**.

Correcting errors

Eclipse allows you to correct errors in the source file. The Debug perspective provides an editor area where the file is opened for debugging. Any other files opened in the editor before debugging, are also opened in the debug perspective. You can modify the source files, save them, build them, and restart the debug session.

Viewing Assembly Language and CPU Registers

Eclipse provides a Disassembly view where you can view the loaded program as assembler instructions mixed with source code.

For more information, navigate to the following page in the online help page for Eclipse.

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **Debug views** → **Disassembly view**.

You can use the Registers view to view the registers used during debugging .

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **Debug views** → **Registers view**.

Debugging Threaded Applications

The debug view of Eclipse lists all the threads of the application being debugged. It displays the stack frames for the suspended threads. The Debug view displays the debugging information in a tree hierarchy. Each thread in the program appears as a node in the tree.

For more information, see the Eclipse online Help.

Attaching a Debugger to a Running Program

Similar to SoftBench, HP-UX DEE also allows you to attach the debugger to a running remote application.

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **HP-UX Development Environment for Eclipse** → **Tasks** → **Attach debugger to a running process**.

Customizing the Debug Perspective

Eclipse allows you to customize the Debug perspective. You can select the views you require during debugging. For more information, see [Customizing your Environment](#) (page 23).

6 Using the Static Analyzer

The SoftBench Static Analyzer helps you to analyze program information such as call trees, class hierarchy, file include relationships, and variable and procedure definitions. It uses the Static database generated during the build of the project by using the `-y` compile option.

The SoftBench Static Analyzer provides the **Show** menu and the **Symbol** menu to execute textual static queries in the project. General queries, such as displaying source files, functions, global variables, and so on, in the current analysis file set can be executed using the **Show** menu. Textual queries about specific symbols such as a specific class or a specific function can be executed using the **Symbol** menu. The **History** menu of the SoftBench Static Analyzer contains the results of successful queries.

HP-UX DEE provides the following alternatives to perform the same tasks as the **Show** and **Symbol** menu of the SoftBench Static analyzer:

- Search dialog
- Context menu options of the C/C++ editor

HP-UX DEE includes code advisor that performs the function of static analysis of the C/C++ code to locate programming errors. For more information, see [Using Code Advise](#).

Search Dialog

The Search dialog enables you to search elements, such as C++ classes, functions, enumerators, structures, and so on in the files of the project. You can also search for files with different extensions in the project.

To invoke the Search dialog, select **Search** → **C/C++...**. The C/C++ Search page of the Search dialog appears. If you select **Search** → **File...**, the File Search page of the Search dialog appears.

Figure 6-1 shows the C/C++ Search page selected in the Search dialog.

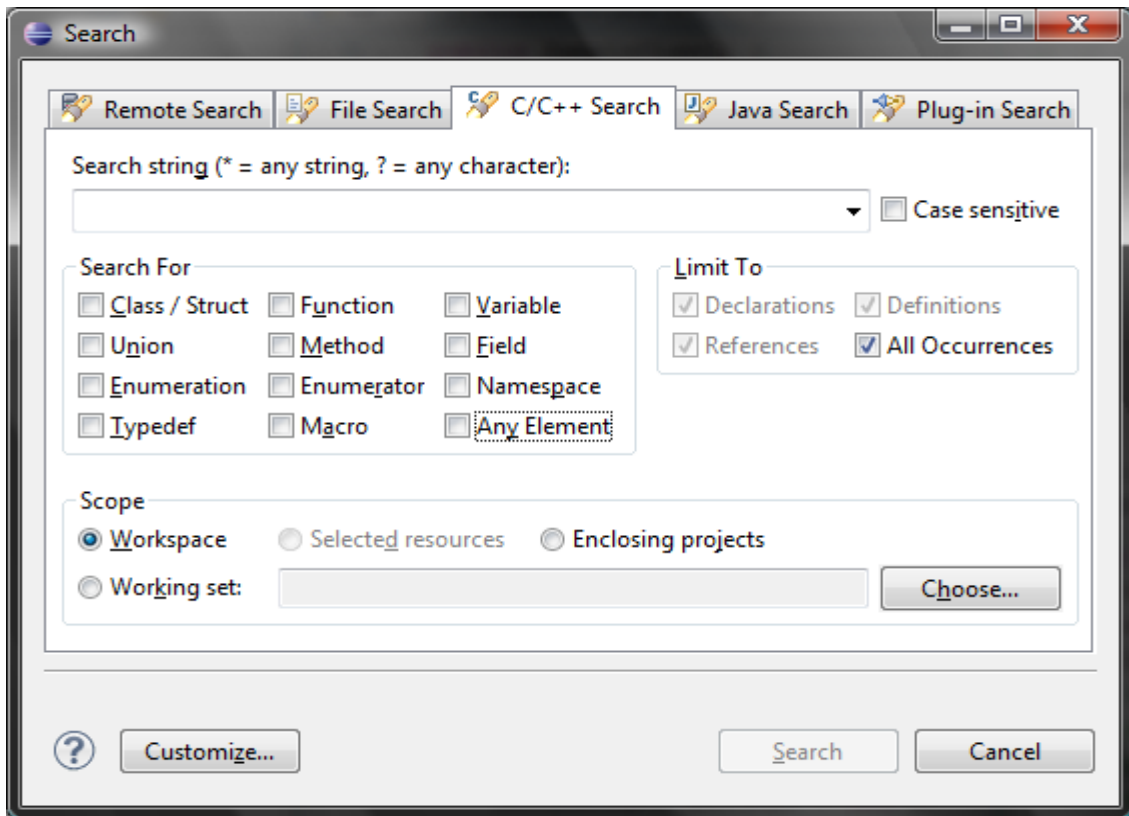


Fig 6-1: C/C++ Search page of the Search dialog

In the C/C++ Search page of the Search dialog:

- Enter the string you want to search in the **Search string** box.
- Select the element you want to search in the **Search For** group.
- Select the type of occurrence you want to search in the **Limit To** group.
- The **Scope** group enables you to specify where the search needs to be performed-in the entire workspace or the selected working set and so on.

To search source files in the project, select the **File Search** tab in the Search dialog, as shown in Figure 6-2.

To search a file containing a particular string, enter the string in the **Containing text:** text box. To search the source files in the project, leave the field **Containing text:** empty.

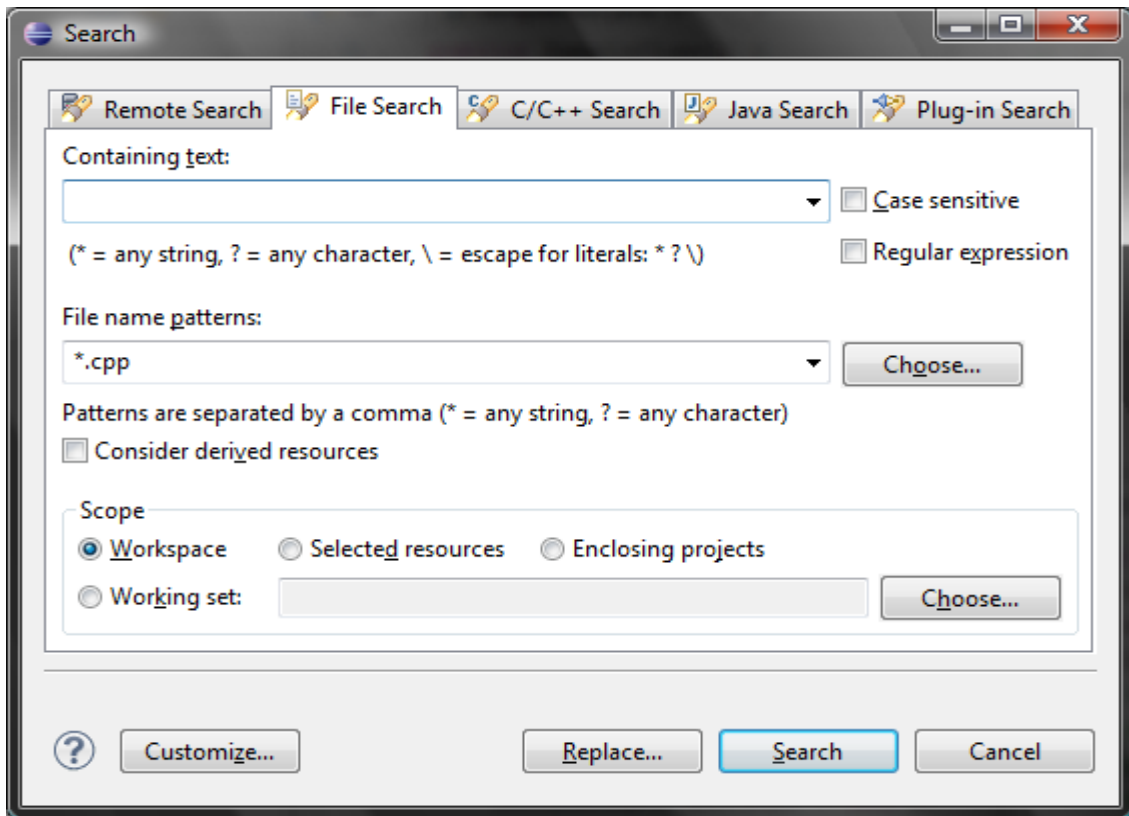


Fig 6-2: File Search dialog

For more information about the Search dialog, navigate to the following page in the online help page for Eclipse:
 In the Contents column, select:

- **C/C++ Development User Guide** → **Concepts** → **C/C++ search**.
- **C/C++ Development User Guide** → **Tasks** → **Searching the CDT**.

Search View

The result of the Search dialog is displayed in the Search View, as shown in Figure 6-3.

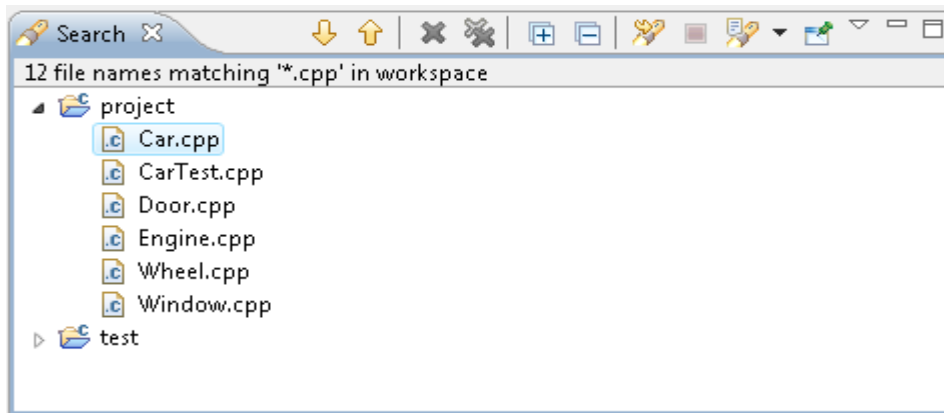


Fig 6-3: Search View

When you search for elements within a file, the Search view displays the filename and line number of the occurrence of the element, shown in Figure 6-4.

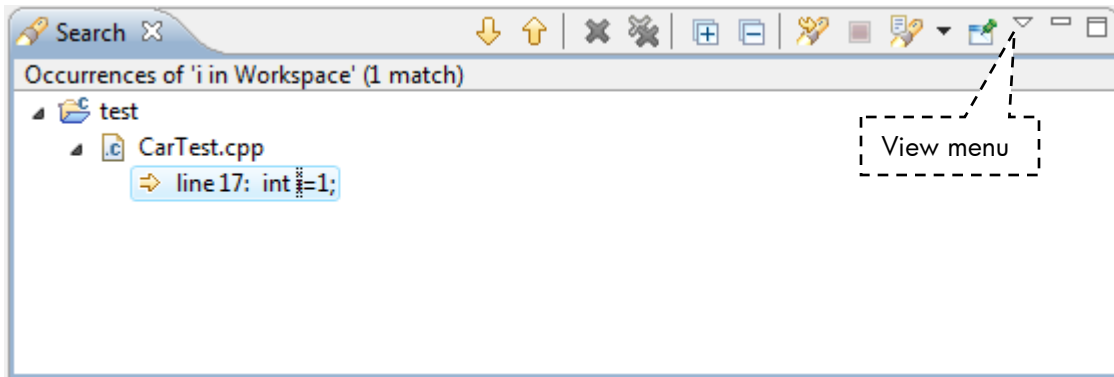


Fig 6-4: Search View with filename and line number of the Search result

You can choose to display the Search results as a 'Tree' or as a 'List' using one of the Search View Toolbar options named **View menu**. The **View menu** icon is shown in Figure 6-4.

Figure 6-5 shows the **View menu**, with the **Show as Tree** option selected. The Search results shown in the figure 6-5 are displayed in Tree format.

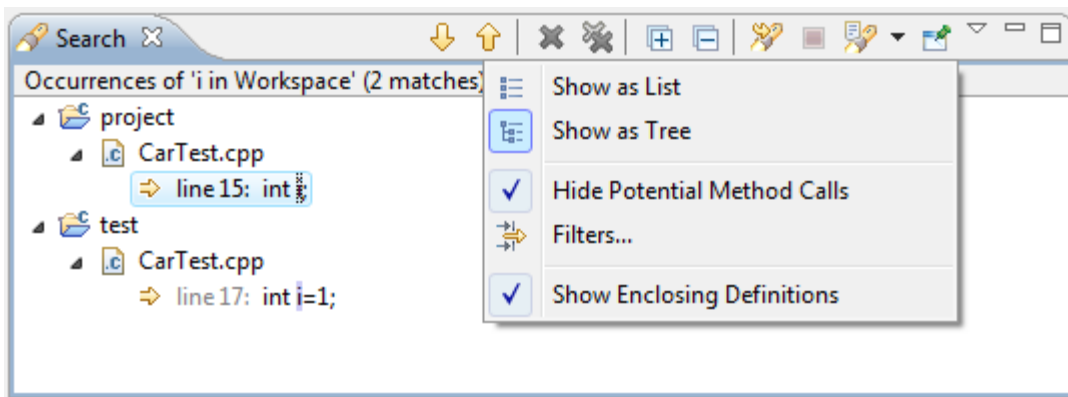


Fig 6-5: Search View with the View menu

Figure 6-6 shows the Search results displayed in List format.

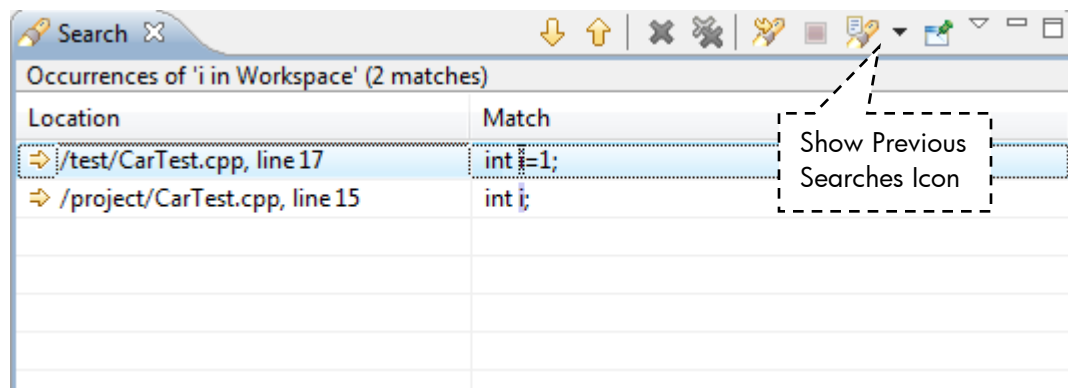


Fig 6-6: Search View with the search results in List form

The Search view toolbar includes the **Show Previous Searches** option, as shown in Figure 6-6. Use this option to view earlier search results.

Clicking the **Show Previous Searches** option displays the **Previous Searches** dialog, as shown in Figure 6-7. In this dialog, select a search query and click **Open** to view the search result, and click **Remove** to delete the

search entry. The Previous Searches dialog includes the **Configure...** option, which enables you to select the number of search queries to be listed in the **Previous Searches** dialog.

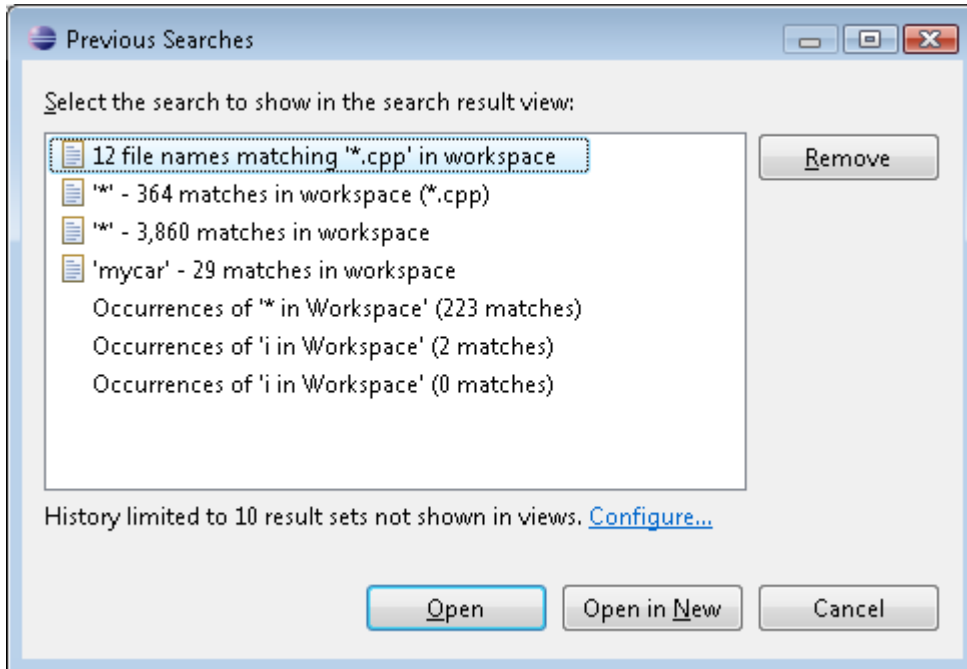


Fig 6-7: Previous Searches Dialog

For information regarding the Search view and the toolbar options of the Search view, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **C/C++ Development User Guide** → **Reference** → **C/C++ Views and Editors** → **Search view**.

Context Menu options of the C/C++ Editor

In Eclipse, some of the functions of the static analyzer can be performed through the editor. To do so, open the file in the editor and select the variable name or the procedure name. From the context menu, select one of the following:

1. Open Declaration: Opens the file where the selected variable or procedure is declared.
2. Open Type Hierarchy: Opens the hierarchy view of the chosen variable or procedure. For more information, navigate to the following page in the online help page for Eclipse. In the contents column, select **C/C++ Development User Guide** → **Reference** → **C/C++ Views and editors** → **Type hierarchy view**.
3. Open Call Hierarchy: Opens the call hierarchy view, which displays the loaded callers and callees for the selected procedure. For more information, navigate to the following page in the online help page for Eclipse. In the contents column, select **C/C++ Development User Guide** → **Reference** → **C/C++ Views and editors** → **Call hierarchy view**. The Open Call Hierarchy feature of Eclipse is similar to the Call Graph of SoftBench, which is one of the four Static graphs.
4. Declarations: Opens in the search menu where the selected variable or procedure is declared.
5. References: Opens in the search menu where the selected variable or procedure is referenced.

The above-mentioned Context menu actions of HP-UX DEE are equivalent to the tasks performed by the menu options **Symbol:References()**, **Symbol:Declarations()** and **Symbol:Definition()** of SoftBench Static Analyzer.

7 Using File Compare

The File Compare feature of SoftBench is available in HP-UX DEE and can be used to compare files and merge files into another file.

Comparing Files

The SoftBench File Compare Tool allows you to compare two files. You can invoke the tool directly without selecting the files to be compared or invoke the tool after selecting the files to be compared. The selected files can belong to the same project, different projects, or not belong to any project.

In Eclipse, to compare files, select the two files to be compared. From the context menu, select **Compare With** and then **Each Other** from the submenu. The two files you select for comparison can belong to the same project or different projects, which are currently in the Eclipse workspace. In Eclipse, you can compare different files; you can compare versions in the Workbench with versions in the repository, or compare a file with its local edit history. This capability of comparing a file with its local edit history helps you to understand the progression of changes that are made to a file. The Local History of a file is maintained when the file is edited and saved. The Local History is displayed in the History View.

For information about the local history and its usage, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **Workbench User Guide** → **Reference** → **User interface information** → **Development environment** → **Local history**.

In some cases, you can compare three files (when a common ancestor exists).

For information about comparing three files, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **Workbench User Guide** → **Concepts** → **Team programming with CVS** → **Three way comparisons**.

If one of the files you select for comparing does not belong to any project in the workspace, create a project that has the required file, and then use the file.

After the comparison, the compare editor opens in the editor area (Figure 7-1).

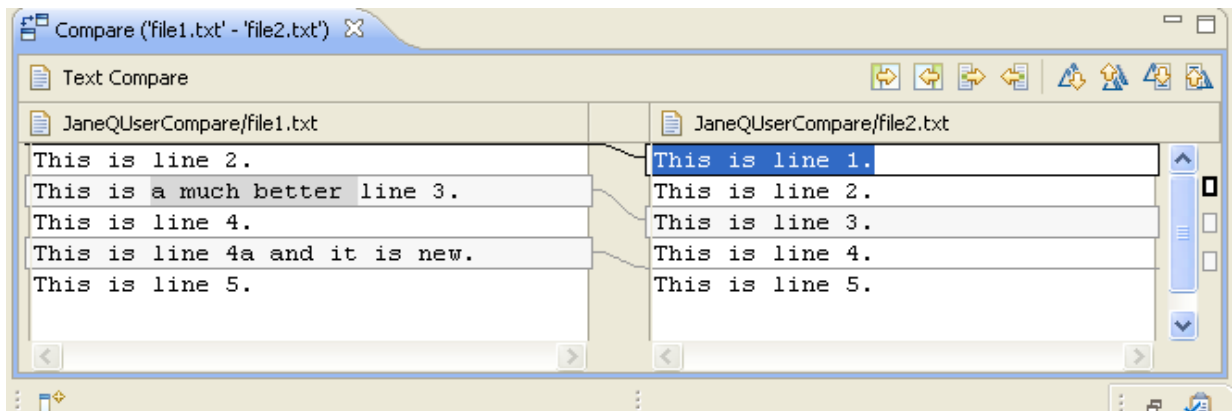


Fig 7-1: Comparison of two source files

Sections of the file that are not identical are highlighted in boxes. The difference in content of each section is highlighted in gray. The Compare editor enables two kind of navigation:

Differences: A section of the file that has one or more lines, and contains many changes.

Changes: A portion of text that is modified within the line.

The toolbar of the Compare editor (located on the top right of the editor) enables you to navigate through the Compare editor and provides options to merge the two files.

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **Workbench User Guide** → **Reference** → **User interface information** → **Views and editors** → **Compare editor**.

Merging Files

The SoftBench File Compare tool allows you to merge the compared files into a new single file. The tool allows you to select the lines from which files should be written to the merged file. To edit the files, open the file using the pre-configured editor.

The toolbar of the Compare editor in Eclipse includes buttons that allow you to merge changes from left file to the right file and vice versa. The following are the types of merges:

1. Copy All from Left to Right
2. Copy All from Right to Left
3. Copy Current Change from Left to Right
4. Copy Current Change from Right to Left

The Compare editor also allows you to edit and save changes to any of the two files open in the Compare editor.

For more information, navigate to the following page in the online help page for Eclipse:

In the Contents column, select **Workbench User Guide** → **Tasks** → **Comparing resources** → **Merging changes in the compare editor**.

8 Unsupported features

The following SoftBench features are not supported in HP-UX DEE:

1. SoftBench Class Graph/Editor
2. SoftBench Configuration Management
3. SoftBench Debugger Data Graph Window
4. Static Graphs
5. Message Monitor
6. SQL related features
7. The Target Graph
8. Customizing by setting X-Resources

9 Support and Other Resources

Intended Audience

This document is intended for programmers who want to migrate from the SoftBench Integrated Development Environment to the HP-UX Development Environment for Eclipse.

Conventions

Table 9-1: Eclipse Terminology

Terms	SoftBench	HP-UX Development Environment for Eclipse
Local Workspace	Local Workspace Root: The root directory of the file system hierarchy where the source files are located, and where they are edited and built. It is not mandatory for your project to include all the files under the hierarchy (you can use these files in different projects).	Eclipse Workspace: The directory in the local system where all the Eclipse projects are stored.
Online help contents	To read the online help for SoftBench, click Help in the SoftBench main window or click Help in the tools' window.	To read the Eclipse online help, click Help in the Eclipse workspace and select Help Contents .
Context menu	-	This menu appears when you right-click a resource.
Mapped Network Drive	-	A local drive on Windows, which can be associated with a shared storage area on another system over the network.
Workbench	-	The desktop development environment. Each Workbench window can have several perspectives.
Perspective	-	The initial set and layout of views in the Workbench window. Each perspective provides a set of functionalities aimed at accomplishing tasks using specific resources.

References

1. *HP-UX DEE Installation Guide* - http://h21007.www2.hp.com/portal/download/files/unprot/hpc/eclipse-plugin/4_0_0/RDP%20Installation%20Guide.pdf
2. *HP-UX DEE User Guide* - http://h21007.www2.hp.com/portal/download/files/unprot/hpc/eclipse-plugin/4_0_0/RDP%20User%20Guide.pdf
3. *HP-UX DEE FAQ* - http://s/h21007.www2.hp.com/portal/download/files/unprot/hpc/eclipse-plugin/4_0_0/RDP_FAQ.pdf
4. Eclipse Online help contents page

Index

B

Breakpoints, 35
build, 20
build order dependency, 11

C

C/C++ Search, 39
cadvise, 31
cloning projects, 10
Code Advisor, 31
Compile Mode, 21
content assist, 26
Context menu, 46
customize, 24, 37

D

Debug Configuration, 32
Debug view, 33
Disassembly view, 37

E

Eclipse, 5
Eclipse editors, 26
Expressions view, 34
External Build, 10

F

File Compare, 43
File Creation, 16
File Search, 39

G

Graph, 42, 45

H

Header file, 19

I

Installation, 6

M

Mapped Network Drive, 46
merge, 44

O

online help, 25

P

Perspective, 7, 46
Preferences, 28
Problems View, 22
Project Build, 9

R

registers view, 37
Remote Standard Make project, 10
repartition a project, 11

S

Search, 38
Signals view, 36
Source file, 18
static analysis, 38
Subprojects, 11

T

Target, 20

V

Variables view, 34

W

watchpoint, 36
Workspace, 46