



# Micro Focus RM/COBOL

CodeWatch User's Guide

Micro Focus  
The Lawn  
22-30 Old Bath Road  
Newbury, Berkshire RG14 1QN  
UK  
<http://www.microfocus.com>

© Copyright 2017-2021 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Revised 2021-04-27 for version 12.18

# Contents

<b>Preface</b> .....	<b>1</b>
Welcome to CodeWatch .....	1
Prerequisites.....	1
About Your Documentation.....	2
Related Publications.....	3
Symbols and Conventions.....	3
Technical Support .....	5
Support Guidelines .....	5
Test Cases .....	5
<b>Chapter 1: Installation and System Considerations</b> .....	<b>7</b>
System Requirements.....	7
Required Hardware .....	7
Required Software .....	8
System Installation.....	8
High-DPI Display Considerations .....	9
<b>Chapter 2: Getting Started with CodeWatch</b> .....	<b>11</b>
Quick Start Tutorial .....	11
Starting CodeWatch.....	17
Starting CodeWatch from Windows .....	17
Starting CodeWatch Using a Saved Workspace .....	17
Starting CodeWatch from a Command Prompt .....	18
Starting CodeWatch with a Runtime Command Line.....	19
Exercise 1—Start CodeWatch .....	19
Configuring a Workspace .....	20
Exercise 2—Configure a Workspace with the Workspace Wizard .....	21
Viewing and Editing Program Source.....	24
Exercise 3—Locate Source Files .....	25
Running the Program .....	25
Exercise 4—Run the Program .....	26
Animating the Program.....	28
Exercise 5—Animate the Program .....	28
Controlling the Flow of Execution.....	28
Setting Breakpoints.....	29
Exercise 6—Set a Breakpoint .....	29
Setting Watchpoints.....	30
Exercise 7—Set a Watchpoint .....	30
Inspecting and Modifying Data.....	31
Exercise 8—Select a Data Item .....	31
Stopping the Program .....	32
Exercise 9—Stop the Program.....	32

Changing the Workspace .....	33
Saving the Workspace.....	33
<b>Chapter 3: CodeWatch Interface.....</b>	<b>35</b>
CodeWatch Workspace Wizard .....	35
Introduction (New) Page.....	36
Set Program to Run Page.....	38
Set Additional Options Page.....	40
Set Environment Page.....	41
CodeWatch Main Window .....	43
Dockable Windows.....	46
Workspace Window.....	47
Programs Tab.....	48
Files Tab .....	49
Context Menu in a Workspace Window.....	49
Data Window .....	51
Breakpoints Tab.....	52
Watchpoints Tab.....	53
Data Files Tab.....	54
Call Stack Tab .....	55
Build Results Tab .....	56
Context Menu in a Data Window .....	57
Log Window .....	58
Context Menu in a Log Window .....	58
Document Windows.....	59
Using Document Windows .....	59
State of Document Windows .....	59
Document Menu .....	60
Cycling Through Open Windows .....	61
Splitting Document Windows.....	61
Closing Document Windows.....	62
Floating Windows.....	62
Find Dialog Box.....	63
Regular Expressions .....	64
Replace Dialog Box .....	65
Inspect/Modify Dialog Box .....	66
Menus and Toolbars.....	68
File Menu.....	69
Edit Menu .....	74
View Menu .....	75
Build Menu .....	76
Run Menu .....	77
Window Menu .....	79
Workspace Properties Dialog Box.....	80
Program Tab .....	81
Options Tab.....	83
Environment Tab .....	85
Command Tab.....	87
Breakpoints & Watchpoints Dialog Box.....	87
Code Breakpoints Tab.....	88
Data Watchpoints Tab.....	89
Item Properties Dialog Box .....	91
Source File Property Tab .....	92
Programs Property Tab .....	93
Build Options Property Tab.....	93

Select Character Set Dialog Box.....	95
<b>Chapter 4: Source Code Editor.....</b>	<b>97</b>
Locating Source Files During Execution .....	97
Source Code/Edit Window .....	97
Contents of the Source Code/Edit Window .....	98
Context Menu for the Source Code/Edit Window .....	99
Using the Source Code/Edit Window .....	101
Creating a New Source Code Document .....	101
Opening an Existing Source Code File .....	102
Viewing Source During Program Execution.....	103
Editing Operations in the Source Code/Edit Window .....	103
Character Sets .....	104
The Workspace Character Set.....	106
Choosing an Editor Font.....	107
Setting Preferences .....	109
Tabs Tab .....	109
Editor Preferences Tab.....	110
External Editing Tab.....	111
Miscellaneous Tab .....	114
Basic Editing Techniques .....	115
Moving Around in a File.....	115
Insertion Point .....	116
How to Move with the Mouse .....	116
How to Move with the Keyboard .....	117
Selecting Text .....	118
How to Select Text with the Mouse.....	118
How to Select Text with the Keyboard.....	120
Inserting New Text .....	120
Overwriting or Replacing Text .....	121
Overwriting Selected Text.....	121
Deleting Text .....	121
Correcting and Editing Text.....	121
Cutting, Copying, and Pasting Text.....	123
Using the Windows Clipboard to Duplicate or Move Text.....	123
<b>Chapter 5: Building Programs with CodeWatch .....</b>	<b>125</b>
Compiling .....	125
Compiling Only Changed Files.....	125
Setting Compilation Options.....	126
The RMPATH and Environment Variables .....	128
Using 32-Bit and 64-Bit Compilers .....	129
Locating the Compiler .....	129
Compiler Output .....	130
<b>Appendix A: Additional Configuration Options .....</b>	<b>131</b>
Runtime Options.....	131
Setting the RM/COBOL Runtime Registry Key.....	131
Setting the RM/COBOL Runtime Configuration .exe File.....	131
Selecting a Runtime License Vault.....	132
Compiler Options.....	132
Setting the RM/COBOL Compiler Registry Key .....	132
Setting the RM/COBOL Compiler Configuration .exe File.....	133
Choosing the 32-bit or 64-bit RM/COBOL Compiler .....	133

Miscellaneous Options.....	134
Changing the Log Window Font.....	134
Setting the Log Window Size .....	134
Controlling Dependency Checking.....	134
<b>Appendix B: Summary of Enhancements.....</b>	<b>135</b>
Version 12 Enhancements.....	135
Version 12.12 Enhancements.....	136
Version 12.14 Enhancements.....	136
Version 11 Enhancements.....	137
Version 10 Enhancements.....	137
Version 9 Enhancements.....	137
Version 8 Enhancements.....	138
Version 7.5 Enhancements.....	139
<b>Index .....</b>	<b>141</b>

**List of Figures**

Figure 1: Glyphs in Western (ANSI) Character Set in “Courier New” Font .....	105
Figure 2: Glyphs in OEM Character Set in “Terminal” Font.....	108

**List of Tables**

Table 1: Types of Workspace Windows .....	46
Table 2: Special Characters Allowed in CodeWatch Regular Expressions .....	64
Table 3: Direction Keys.....	117
Table 4: Text Selection Keys.....	120
Table 5: Editing Keys .....	121
Table 6: Edit Menu Shortcut Keys.....	123
Table 7: RM/COBOL Compile Command Options Available .....	127

# Preface

---

## Welcome to CodeWatch

This document describes CodeWatch, Micro Focus' fully integrated development environment for RM/COBOL for Windows.

CodeWatch for Windows supports the entire development cycle, including editing, compiling, and debugging of RM/COBOL applications. CodeWatch can be used to debug and change programs that are independently compiled, without requiring you to build projects. Instead, all the knowledge about the structure of your application is built up during debugging sessions.

The new features for the most recent release of CodeWatch are described in [Appendix B: Summary of Enhancements](#) (on page 135).

**Note** CodeWatch version 12 is now able to debug service programs running under Micro Focus' Xcentrinity Business Information Server (BIS) on Microsoft Windows under Internet Information Server (IIS). For more information, see the *CodeWatch for Xcentrinity Business Information Server User's Guide Supplement*.

---

## Prerequisites

Several conditions are necessary before you can run CodeWatch for Windows:

- CodeWatch requires that the following products be installed on your system:
  - One of the following operating systems is required:
    - Microsoft Windows XP
    - Microsoft Windows Server 2003
    - Microsoft Windows Vista
    - Microsoft Windows Server 2008
    - Microsoft Windows Server 2008 R2
    - Microsoft Windows 7

- Microsoft Windows 8 and 8.1
  - Microsoft Windows Server 2012
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows 10
- Microsoft Windows Server 2016RM/COBOL for Windows development system, version 12.
- For source-level debugging, the COBOL object files must be compiled using the RM/COBOL version 7.5 or later compiler (note that version 10 or later is strongly recommended), using the Full Debug Information (Y=2 or Y=3) Compile Command Option (see Chapter 6: *Compiling* of the *RM/COBOL User's Guide*). This option embeds additional symbol information in the object file. A new option in the **rmpgmcom** utility, which is provided with the development system, can be used to strip this information from object files without the need to recompile (see Appendix G: *Utilities* of the *RM/COBOL User's Guide*).
  - For non-source level debugging, a printed RM/COBOL compiler listing file (produced by the compiler when the program was originally compiled) is required. CodeWatch can be used to set breakpoints and watch variables during non-source level debugging.
  - To debug programs that use the 64-bit RM/COBOL runtime, CodeWatch version 12.13 (or later) is required.
  - To compile programs using the 64-bit RM/COBOL compiler, CodeWatch version 12.14 (or later) is required.

---

## About Your Documentation

The CodeWatch documentation consists of a user's guide, which is available in Portable Document Format (PDF) on the product CD, and on the web at <https://supportline.microfocus.com/productdoc.aspx>, Micro Focus Developer – COBOL and Software Developer Tools | RM/COBOL | 12 | CodeWatch User's Guide.

**Note** Viewing a PDF file requires Adobe Acrobat Reader. You can download the latest version of Adobe Acrobat Reader free at the Adobe website at <http://www.adobe.com>.

This user's guide is divided into the following parts:

**Chapter 1—Installation and System Considerations** lists the hardware and software required to install CodeWatch for Windows, and describes how to install the product.

**Chapter 2—Getting Started with CodeWatch** provides a quick start tutorial of CodeWatch followed by a more in-depth, step-by-step walk through of a basic CodeWatch debugging session.

**Chapter 3—CodeWatch Interface** provides detailed descriptions of the Workspace Wizard, and the various windows, dialog boxes, menus, and toolbars found in CodeWatch.

**Chapter 4—Source Code Editor** explains how CodeWatch locates source files, describes the Source Code/Edit window and how to use the internal Source Code Editor, discusses setting preferences to customize the editing environment (including configuring an external editor), and provides an overview of the most common and basic editing tasks in CodeWatch.

**Chapter 5—Building Programs with CodeWatch** provides an overview of how CodeWatch compiles source files.

**Appendix A—Additional Configuration Options** describes a file called **rmcw.ini**, which is used by CodeWatch to include several optional configuration options and tuning parameters that are not set from within the program.

**Appendix B—Summary of Enhancements** provides an overview of the new features in the current release, and reviews the changes and enhancements that were added to earlier releases of CodeWatch.

The *CodeWatch User's Guide* also includes an index.

---

## Related Publications

For additional information, refer to the following publications:

*CodeBridge User's Guide*

*CodeWatch for Xcentrity Business Information Server User's Guide Supplement*

*RM/COBOL User's Guide*

*RM/COBOL Language Reference Manual*

*RM/COBOL Syntax Summary Help File*

*WOW Extensions Designer Help File and WOW Extensions Functions and Messages Help File*

*Xcentrity Business Information Server (BIS) User's Guide*

*XML Extensions User's Guide*

---

## Symbols and Conventions

The following typographic conventions are used throughout this manual to help you understand the text material and to define syntax:

1. Words in all capital letters indicate COBOL reserved words, such as statements, phrases, and clauses; acronyms; configuration keywords; environment variables, and RM/COBOL Compiler and Runtime Command line options.
2. Text that is displayed in a monospaced font indicates user input or system output (according to context as it appears on the screen). This type style is also used for sample command lines, program code and file listing examples, and sample sessions.

3. Bold, lowercase letters represent filenames, directory names, and programs.

Words you are instructed to type appear in bold. Bold type style is also used for emphasis, generally in some types of lists.

4. Italic type identifies the titles of other books and names of chapters in this guide, and it is also used occasionally for emphasis.

In syntax, italic text denotes a placeholder or variable for information you supply, as described below.

5. The symbols found in the syntax charts are used as follows:
  - a. *italicized words* indicate items for which you substitute a specific value.

- b. UPPERCASE WORDS indicate items that you enter exactly as shown (although not necessarily in uppercase).
  - c. ... indicates indefinite repetition of the last item.
  - d. | separates alternatives (an either/or choice).
  - e. [ ] enclose optional items or parameters.
  - f. { } enclose a set of alternatives, one of which is required.
  - g. {} surround a set of unique alternatives, one or more of which is required, but each alternative may be specified only once; when multiple alternatives are specified, they may be specified in any order.
6. All punctuation must appear exactly as shown.
  7. Key combinations, sometimes called shortcut keys, are connected by a plus sign (+), for example, Ctrl+X. This notation indicates that you press and hold down the first key while you press the second key. For example, “press Ctrl+X” means to press and hold down the Ctrl key while pressing the X key. Then release both keys.
  8. Note the distinction of the following terminology:
    - The term “window” refers to a delineated area of the screen, normally smaller than the full screen.
    - The term “Windows” refers to Microsoft 32-bit Windows operating systems, including Windows 2000, Windows XP, or Windows Server 2003, Windows Vista, or Windows Server 2008, unless specifically stated otherwise.

#### Note

RM/COBOL and CodeWatch no longer support earlier Microsoft Windows operating systems, including Microsoft Windows 98, Windows 98 SE, Windows Me, and Windows NT 4.0

Furthermore, any references to these versions, or to the shorthand notation “Windows 9x class” or “Windows NT class” that refer to these operating systems, are included for historical purposes only.

9. RM/COBOL Compile and Runtime Command line options may be preceded by a hyphen. If any option is preceded by a hyphen, then a leading hyphen must precede all options. When assigning a value to an option, the equal sign is optional if leading hyphens are used.
10. In the electronic PDF file, this symbol represents a “note” that allows you to view last-minute comments about a specific topic on the page in which it occurs. This same information is also contained in the README file under the section, *Documentation Changes*. In Adobe Reader, you can open comments and review their contents, although you cannot edit the comments. Notes do not print directly from the comment that they annotate. You may, however, copy and paste the comment text into another application, such as Microsoft Word, if you wish.



To review notes, do one of the following:

- To view a note, position the mouse over the note icon until the note description pops up.
- To open a note, double-click the note icon.
- To close a note, click the Close box in the upper-left corner of the note window.

---

# Technical Support

Micro Focus is dedicated to helping you achieve the highest possible performance from the Micro Focus RM/COBOL family of products. The Micro Focus Customer Care team is committed to providing you with prompt and professional service when you have problems or questions about your Micro Focus products.

These technical support services are subject to Micro Focus' prices, terms, and conditions in place at the time the service is requested.

While it is not possible to maintain and support specific releases of all software indefinitely, we offer priority support for the most current release of each product. For customers who elect not to upgrade to the most current release of the products, support is provided on a limited basis, as time and resources allow.

## Support Guidelines

When you need assistance, you can expedite your call by having the following information available for the technical support representative:

1. Company name and contact information.
2. Micro Focus RM/COBOL product serial number (found in the Electronic Software Delivery email, on the media label, or in the product banner message).
3. Micro Focus RM/COBOL product version number.
4. Operating system and version number.
5. Hardware, related equipment, and terminal type.
6. Exact message appearing on screen.
7. Concise explanation of the problem and process involved when the problem occurred.

## Test Cases

You may be asked for an example (test case) that demonstrates the problem. Please remember the following guidelines when submitting a test case:

- The smaller the test case is, the faster we will be able to isolate the cause of the problem.
- Do not send full applications.
- Reduce the test case to the smallest possible combination of components required to reproduce the problem.
- If you have very large data files, write a small program to read in your current data files and to create new data files with as few records as necessary to reproduce the problem.
- Test the test case before sending it to us to ensure that you have included all the necessary components to recompile and run the test case. You may need to include an RM/COBOL configuration file.

When submitting your test case, please include the following items:

1. **README text file that explains the problems.** This file must include information regarding the hardware, operating system, and versions of all relevant software (including

the operating system and all Micro Focus products). It must also include step-by-step instructions to reproduce the behavior.

2. **Program source files.** We require source for any program that is called during the course of the test case. Be sure to include any copy files necessary for recompilation.
3. **Data files required by the programs.** These files should be as small as possible to reproduce the problem described in the test case.

# Chapter 1: Installation and System Considerations

This chapter lists the hardware and software required to install CodeWatch for Windows and describes how to install the product.

**Note** For information on BIS-based debugging on an IIS web server, see the *CodeWatch for Xcentrinity Business Information Server User's Guide Supplement (Version 12)*.

---

## System Requirements

Your computer configuration is the assembled set of hardware and software that makes up your system. Before you install CodeWatch, please make sure that your computer meets these minimum requirements.

### Required Hardware

To run CodeWatch, you must have hardware similar to what is needed to run RM/COBOL version 12 for Microsoft 32-bit Windows. These system requirements include:

- An IBM PC or compatible machine, capable of running Microsoft Windows 7 SP1 or later, is required.
- A minimum of 1024 megabytes of random access memory (RAM). Depending on your configuration, CodeWatch may run with less memory, but this is not recommended.
- Ten megabytes of disk space for the CodeWatch program and supporting dynamic-link libraries (DLLs), and for updating system libraries, as required.
- An 1024 x 768 x 256 color display adapter. (1280 x 1024 x 256 or better is recommended, however.) Adapters that support more than 256 colors will eliminate screen flashing as you move between programs. Although CodeWatch will run in 640 x 480 x 256, this is not recommended.

**Note** Most Micro Focus RM/COBOL products are distributed electronically. If you elect to receive physical media, an optical drive capable of reading a CD is required at installation time. The license is always distributed electronically.

## Required Software

To run CodeWatch, you must have the following software installed on your computer:

- One of the following operating systems is required:
  - Microsoft Windows 7 SP1
  - Microsoft Windows Server 2008 R2
  - Microsoft Windows 8 and 8.1
  - Microsoft Windows Server 2012
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows 10
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2019

As of version 12.18, Microsoft Windows XP and Microsoft Windows Vista, along with the equivalent servers (2003 and 2008), are no longer supported.

- An RM/COBOL development system, version 12 for Microsoft 32-bit Windows or 64-bit Windows.
  - To run RM/COBOL programs with the 64-bit runtime system requires CodeWatch version 12.13.
  - To compile RM/COBOL programs with the 64-bit compiler requires CodeWatch version 12.14.

---

## System Installation

CodeWatch is generally installed when you install an RM/COBOL for Windows development or runtime system. However, if you did not install CodeWatch when you initially installed RM/COBOL, simply reinstall RM/COBOL and select *only* the CodeWatch component for installation.

You may also automate the installation program for CodeWatch, which allows the installation of CodeWatch to be incorporated with the installation of your application. For more information, navigate to <https://supportline.microfocus.com/productdoc.aspx>, Micro Focus Developer – COBOL and Software Developer Tools | RM/COBOL | 12 | Windows Installer Guide and refer to the “RM/COBOL CodeWatch Integrated Development Environment” section of the *RM-Install Program User's Reference* documentation.

**Installation Note** After any installation of an RM/COBOL development system, the default character set for new workspaces is OEM/DOS. If you wish to use the ANSI/Windows character set as the default for new workspaces, the **RMSETNCS** utility, described in the *RM/COBOL User's Guide*, should be run with the `/cs ansi` command line option. Existing workspaces are not affected by the default character set for new workspaces. For information on changing the character set for a workspace, see the [Select Character Set dialog box](#) (on page 95). For information on character sets in CodeWatch, see [Character Sets](#) (on page 104).

For further details on character sets in RM/COBOL for Windows, including details on the **RMSETNCS** utility, see “Character Set Considerations for Windows” in Chapter 3: *Installation and System Considerations for Windows* of the *RM/COBOL User's Guide*.

---

# High-DPI Display Considerations

Microsoft Windows 8.1 and later support high-DPI displays, which can provide very high resolution and sharper text and graphics. These are displays that have more than 96 dots-per-inch (DPI). For example, a 4K 32" display will typically have 144 DPI, and 4K laptop screens may have even higher pixel density.

For backward-compatibility, Windows 8.1 and 10 support the automatic scaling of applications. The system-wide scale factor is set in the Display settings application. Typical settings are 100% for no scaling, 125% for 120dpi displays, 150% for 144dpi displays, and 175% or even 200% for even higher DPI-displays. This causes the running application to appear at about the same size as on a regular display but the user is free to choose any scale factor. Note that if scaling is enabled, the application is unaware of the user-set scale factor of the system or the actual DPI of the screen, as these values are always reported as 96 DPI with a scale factor of 100%.

Scaling does have a drawback: fonts and other graphical elements may appear slightly fuzzy or blurry as these elements are displayed larger than the size used to render the elements.

Before version 12.18, CodeWatch was never automatically scaled and windows were always rendered at the native display resolution. This included the RM/COBOL runtime system running under CodeWatch for debugging– the runtime inherits the scaling/no-scaling setting used by CodeWatch. Because the runtime system is always scaled by the system when running outside CodeWatch, if any scale factor other than 100% was in effect, there were differences in the appearance of the runtime system when running in and outside of CodeWatch. This could also affect the appearance and layout of graphical WOW programs running under CodeWatch.

Starting with version 12.18, CodeWatch now has a setting that controls whether the application will be scaled or drawn at the native display resolution (the default). See the [Scale Windows on high-DPI Displays](#) setting on the Miscellaneous tab of the Preferences dialog box. A change in the scaling setting requires that CodeWatch be restarted, and the setting is persistent.

Note that, for compatibility with previous versions of CodeWatch, the default setting is to leave automatic scaling disabled and render all windows (including windows created by the runtime system) as if the scale factor was at 100%



# Chapter 2: Getting Started with CodeWatch

This chapter provides a quick start tutorial of CodeWatch. The tutorial is followed by a more in-depth, step-by-step walk through of a basic CodeWatch debugging session, including sections and exercises on:

- [Starting CodeWatch](#) (see page 17)
- [Configuring a workspace](#) (see page 20)
- [Viewing and editing program source](#) (see page 24)
- [Running the program](#) (see page 25)
- [Changing the workspace](#) (see page 33)
- [Saving the workspace](#) (see page 33)

**Note** The information provided in this tutorial section is for traditional CodeWatch for Windows source code debugging. For BIS-based debugging on an IIS web server, see the *CodeWatch for Xcentrivity Business Information Server User's Guide Supplement (Version 12)*.

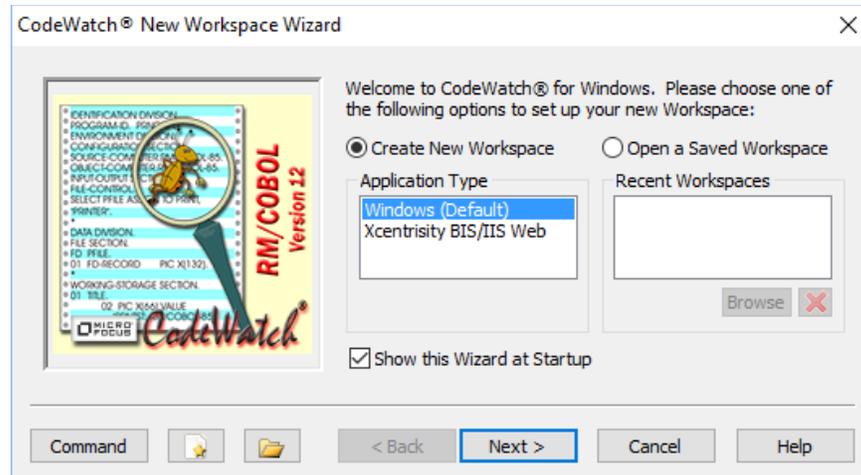
---

## Quick Start Tutorial

This section presents an overview of the process of creating a new program in a basic CodeWatch debugging session. The steps include the following:

1. Start CodeWatch from the CodeWatch desktop icon, or click the Windows **Start** menu, select **Programs, RMCOBOL** and **Development v12**, and then click **CodeWatch**. On Windows 10, click the Windows **Start** menu, select **All Apps, RM Development v12**, and then click **CodeWatch**.

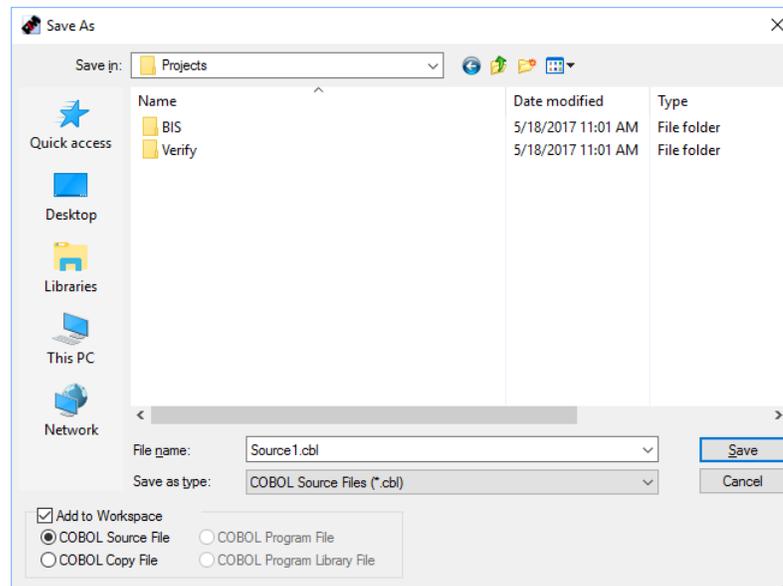
You will see the Workspace wizard, illustrated below. If you do not see the wizard, from the **File** menu, choose **New | Workspace**.



2. In the Workspace wizard, press the **New File** button (depicted to the left). If you prefer, you can dismiss the wizard by pressing the **Cancel** button or the **Esc** key, and then choose **New** from the **File** menu or press **Ctrl+N**.

In either case, a blank page will appear in the CodeWatch main window.

3. From the **File** menu, choose **Save**. The Save As dialog box is displayed.



If you don't see the "Projects" folder as depicted above, navigate to:

```
C:\Users\Public\Documents\Micro Focus\RM\CodeWatch\Projects
```

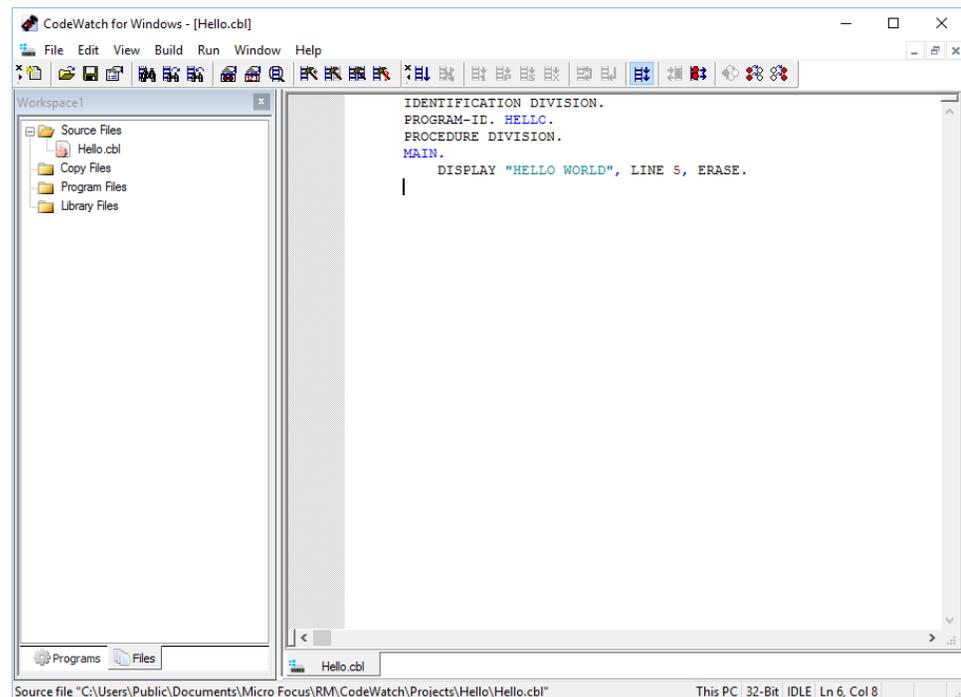
4. Create a folder named "Hello" by clicking the "New Folder" icon on the icon bar. Then double-click on the "Hello" Folder to open it, then type **Hello.cbl** in the **File name** box. Or, choose the directory wherein you wish to work, and then type in **Hello.cbl** in the **File name** box.
5. Make sure that the **Add to Workspace** check box and the **COBOL Source File** option are selected.

6. Press **Enter** or click **Save**.

CodeWatch now knows that you are building a COBOL program and will wrap the source file in a workspace. Within the [CodeWatch main window](#) (see page 43), click in the Source Code/Edit window and enter the following program. Press the Tab key on the first line to move the cursor into column 8. The remaining lines will be indented automatically.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. HELLO.  
PROCEDURE DIVISION.  
MAIN.  
    DISPLAY "HELLO WORLD", LINE 5, ERASE.
```

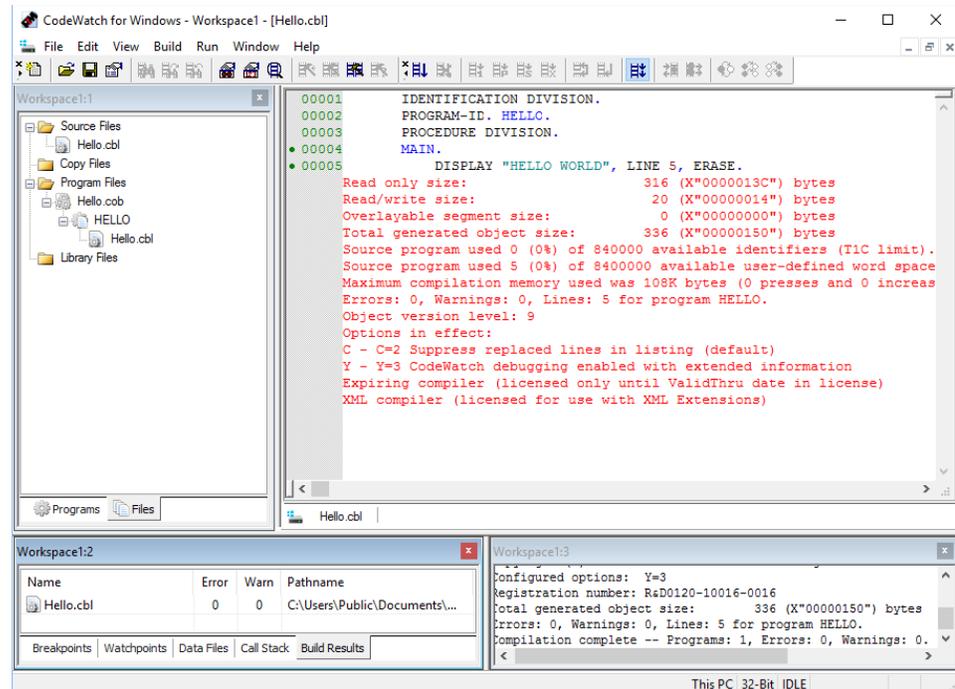
The following figure illustrates your screen.



**Note** The icon for **Hello.cbl** on the Files tab is red, which indicates that the file has not been compiled. Different elements of the program are colored on your screen: COBOL reserved words are black, literal strings are cyan, and user-defined words are blue. The status bar indicates “This PC”, which is the machine being targeted for debugging; “32-Bit”, which is the runtime system that will be used; and “IDLE”, the CodeWatch state. For more information, see [Contents of the Source Code/Edit Window](#) (on page 98).

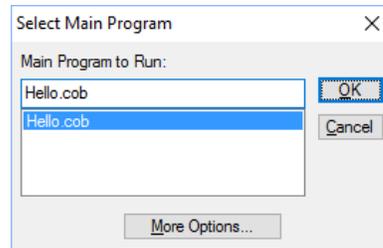
7. Compile the program by clicking in the Source Code/Edit window and choosing **Compile “Hello.cbl”** from the **Build** menu. (If your focus is not in this edit window, choose the **Compile File** menu command.)
8. If you have not saved the program, you will be prompted to save your changes. Press the **Y** key or click **Yes**.

The file will be saved and compiled, and you will see the following in the CodeWatch main window.



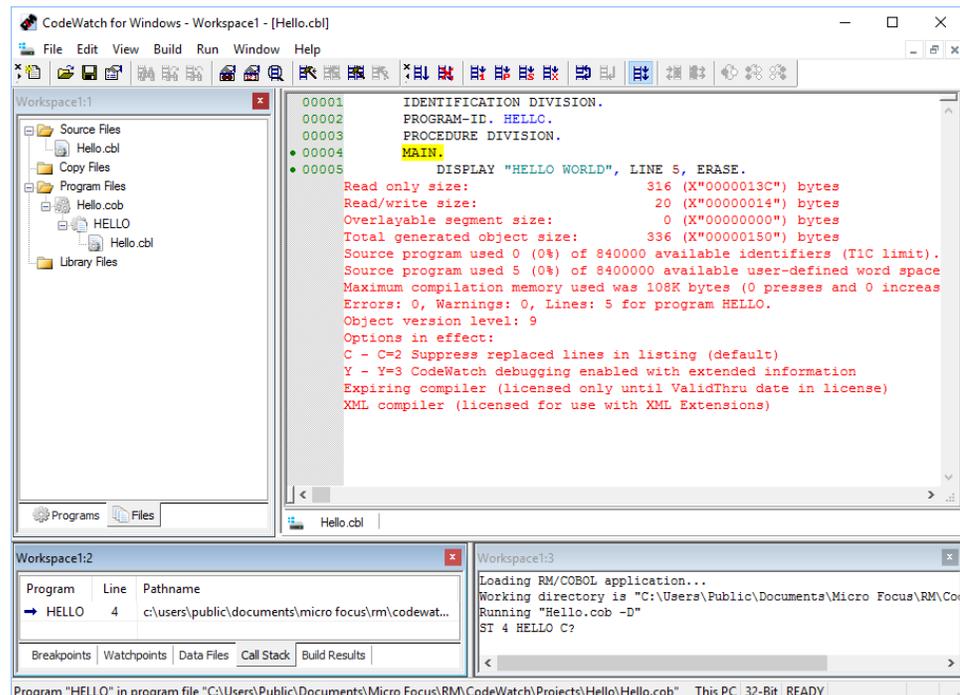
Notice the following changes:

- In the Workspace window in the upper-left corner (Workspace 1:1, in this illustration), the icon to the left of **Hello.cbl** is no longer red, indicating the file is ready to run. A new item, **Hello.cob**, has been inserted. This item represents the generated program file. Expanding the item by clicking the [+ ] reveals a list of programs in the RM/COBOL object file (“HELLO”), and the list of source and COPY files that the object file depends upon (“Hello.cbl” above).
  - In the Source Code/Edit window, line numbers and diagnostic lines have been added to the source lines. A green dot to the left of a line number indicates that the statement is executable. Breakpoints should be placed only on executable statements.
  - The Data window (Workspace 1:2, in this illustration) is now visible in the lower-left corner. This window contains various tabs. The **Build Results** tab has been selected and the window contains the results of the compilation. For more details, see [Data Window](#) (on page 51).
  - The Log window (Workspace 1:3, in this illustration) is visible in the lower-right corner. This window contains the output produced during the compilation.
9. Next, let’s run this program. From the **Run** menu, choose **Start** (or press **F5**). The following prompt will appear in the Select Main Program dialog box the first time this program has been run. Since this workspace contains only one program, only one program is listed.



**Note** Clicking **More Options** opens the Program page of the Workspace Properties dialog box. Refer to [Program Tab](#) (on page 81) for further information.

10. Click **OK**. CodeWatch will remember your choice. The CodeWatch main window will appear as shown below.



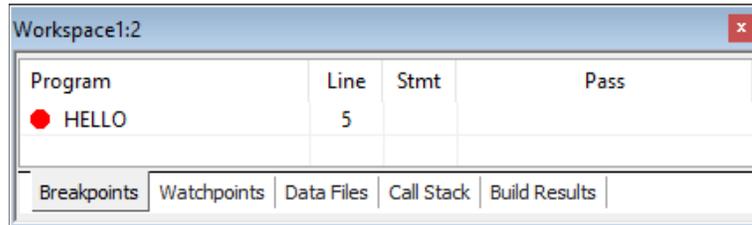
Notice the following changes that have occurred:

- The line that contains the MAIN declaration is highlighted. This is the first statement in the program.
- If the Data window does not display call stack information, click the **Call Stack** tab. This area indicates that you are about to execute line 4 in the program, HELLO. The name of the program file or library is also visible.
- The Log window displays the state of the Debugger. You can click in this window and manually issue RM/COBOL Debug commands, if you so choose.
- The RM/COBOL runtime window has been created and may be partially visible behind the CodeWatch workspace.
- The status bar indicates whether a 32-bit or 64-bit runtime is being used.
- The status bar indicator now indicates “READY.”

11. To set a breakpoint, click anywhere in line 00005 and from the **Run** menu, choose **Breakpoint | Set** or press **F9**.

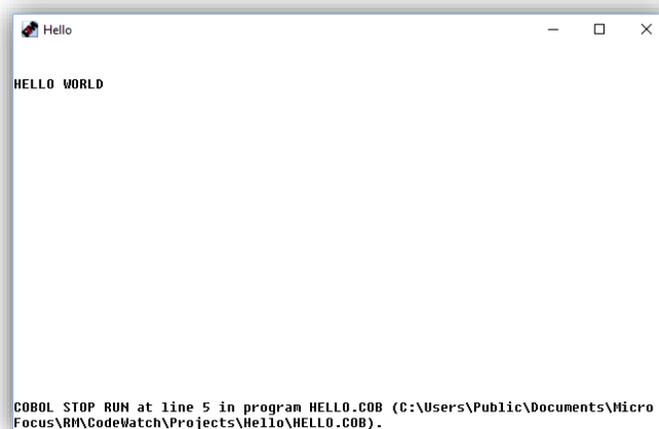
A solid red circle will appear to the left of the line number to indicate that a breakpoint has been set.

12. In the Data window, click the **Breakpoints** tab. This area indicates that execution will stop at line 5 in program HELLO.



13. From the **Run** menu, choose **Resume** or press **F5** to run the program. The highlight will advance to the line containing the breakpoint and execution will stop.
14. To single-step the program, choose **Step Statement** or press **F8** from the **Run** menu.

You should see **HELLO WORLD** in the RM/COBOL runtime window. (If this window is not visible, press **F12** to bring it to the front.)



15. Click anywhere in the CodeWatch main window and press **F8** to step again, or press **F5** to run the program to completion.

The RM/COBOL runtime window will close and CodeWatch will be “IDLE” again.

16. To exit CodeWatch, choose **Exit** from the **File** menu.
17. At the prompt to save your changes to Workspace1, click **Yes** and give the workspace a name (for example, **Hello**).

The workspace will be saved as a workspace file, **Hello.cwf**, and may be quickly reloaded later. See [Starting CodeWatch Using a Saved Workspace](#) (on page 17).

---

## Starting CodeWatch

When you install CodeWatch, the installation process automatically creates a new Start Menu item for CodeWatch in Windows. You have a number of choices for starting CodeWatch, including the following:

- [Starting CodeWatch from Windows](#) (see the next topic)
- [Starting CodeWatch using a saved workspace](#) (see page 17)
- [Starting CodeWatch from a Command Prompt](#) (see page 18)
- [Starting CodeWatch with a Runtime Command line](#) (see page 19)

### Starting CodeWatch from Windows

If you have started and are running Windows, use one of the following methods to start CodeWatch:

- Click the Windows **Start** menu (usually located at the bottom left corner of your screen), point to **RM Development V12** (or **Programs**, point to **RMCOBOL** and then **Development v12**), and then click **CodeWatch**.
- Click the Windows Search button (or choose the **Start** menu and choose **Run**), and then enter **rmcw.exe** and press Enter.
- From Windows Explorer, navigate to the RM/COBOLv12 Program directory (by default, this is **C:\Program Files\Micro Focus\RM\RMCOBOLv12** on Windows 32-bit or **C:\Program Files\Micro Focus\RM\RMCOBOLv12-64** or **C:\Program Files (x86)\Micro Focus\RM\RMCOBOLv12** on Windows 64-bit), and then double-click the CodeWatch icon (**rmcw.exe**) that was created by the installation procedure.
- If you elected to create and put a CodeWatch icon (a shortcut) on your desktop during installation, double-click the icon.

Using these methods, CodeWatch normally opens the Workspace wizard to help you establish and configure a workspace for the session. For more information, see [Configuring a Workspace](#) (on page 20).

### Starting CodeWatch Using a Saved Workspace

You may also start CodeWatch by opening an existing workspace file:

- From Windows Explorer, double-click a saved CodeWatch workspace file (workspace files have an extension of **.cwf**, which represents **C**ode**W**atch **W**orkspace **F**ile).
- Alternatively, right-click the workspace file in Windows Explorer and select **Open** from the pop-up context-sensitive menu. This method allows you to resume debugging using the same set of configuration options, program files, program libraries, and source files that you used when you saved the workspace in a previous CodeWatch session.

**Note** Using Windows Explorer to start a saved workspace requires that the workspace file extension be registered with Windows. If the **.cwf** files are no longer associated with CodeWatch, you can still launch CodeWatch with a saved workspace by dragging and dropping a saved workspace file onto the CodeWatch icon. It may be necessary to use

this technique if you run other applications that also use the **.cwf** file type for document files. Once started, CodeWatch will automatically repair the **.cwf** extension.

- To start CodeWatch with a previously saved CodeWatch workspace, using a command line, enter:

```
rmcw filename.cwf
```

This command causes CodeWatch to open the specified workspace file. The **.cwf** extension must be specified, because it uniquely identifies this command line format.

When you start CodeWatch using a saved workspace file, there are several considerations to keep in mind:

- Workspace files are ASCII text files and can be carefully edited outside of CodeWatch. For additional information, see [External Editing Tab](#) (on page 110).
- To avoid ambiguity, COBOL programs should not use the **.cwf** extension.
- If the filename contains a space, the entire filename must be enclosed in double quotes.

## Starting CodeWatch from a Command Prompt

You may occasionally want to start CodeWatch from a Microsoft DOS command prompt shell window. Starting CodeWatch from a command prompt provides several options:

- To start CodeWatch from a command prompt, enter:

```
start rmcw
```

The Workspace wizard is displayed unless it has been disabled, as described in the following item.

- To start CodeWatch without displaying the Workspace wizard, enter *one* of the following:

```
start rmcw /n
```

```
start rmcw -n
```

- To start CodeWatch with the name of a saved workspace file:

```
start rmcw filename.cwf
```

**Note** To invoke the Workspace wizard at any time, from the **File** menu, choose **New** and then **Workspace**. If a workspace is currently open, do one of the following to invoke the wizard:

- From the **File** menu, choose **Workspace** and then click **New**.
- From the **File** menu, choose **Workspace** and then click **Properties**.

CodeWatch also may be started with an RM/COBOL Runtime Command line, as described in the following section.

**Note** In addition to being able to specify a RUNCOBOL command line, or the name of a CodeWatch workspace file, you can also start CodeWatch from a DOS command line with the name of a **.CBL** or **.CPY** file. CodeWatch will create a default workspace for the file, and set the current working directory to the directory that contains the file.

## Starting CodeWatch with a Runtime Command Line

**Note** CodeWatch command line parameter options may be specified when CodeWatch is started from a Windows command prompt, from Windows 10 Search, from the **Start | Run** menu, or within a Windows shortcut icon.

To start CodeWatch with an RM/COBOL Runtime Command line, enter:

```
start rmcw runcobol-command-line
```

This command creates a workspace using the RM/COBOL runtime options specified in *runcobol-command-line*. Note that *runcobol-command-line* should not include the name of the runtime itself, but may include any **runcobol** options. For example:

```
start rmcw mainprog L=lib\lib1.cob L=lib2.cob C=myconfig.cfg
```

or the equivalent:

```
start rmcw mainprog /L lib\lib1.cob /L lib2.cob /C mycfg.cfg
```

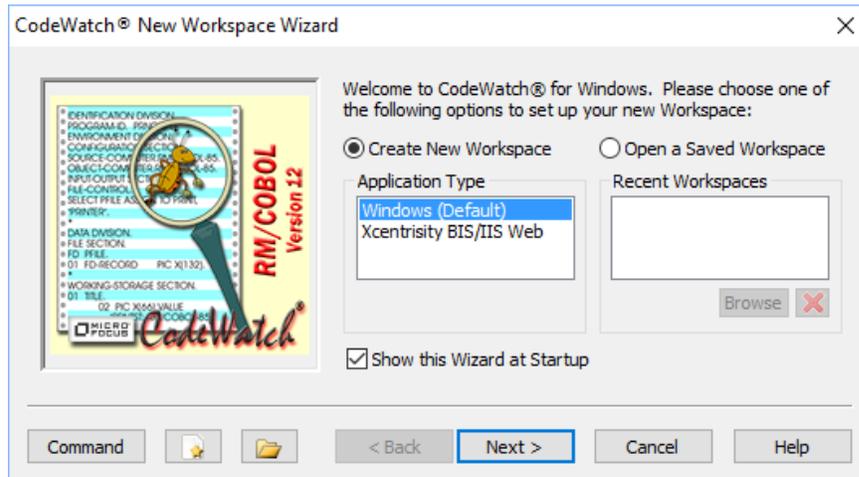
When you start CodeWatch with a Runtime Command line, there are several considerations to keep in mind:

- CodeWatch accepts either the Windows or UNIX style, but the two styles cannot be mixed in a single command. See Chapter 7: *Running of the RM/COBOL User's Guide* for a complete description of the Runtime Command line syntax.
- A dash (-) may be used instead of a slash (/) in the UNIX style, and the option letters are not case-sensitive in either style.
- If a filename contains a space, the entire filename must be enclosed in double quotes.
- The /D (Debug) Runtime Command Option is always assumed to be present, even if not specified. This allows **rmcw** to be used instead of **runcobol** in existing batch scripts. The /D Option may be disabled by choosing **Properties** from the **File** menu and selecting the No Debug option on the [Program tab](#) (see page 81) of the Workspace Properties dialog box.
- As is true with the RM/COBOL runtime system, some options cannot be set from the Windows command prompt when using CodeWatch. It may be necessary to define RMPATH, RUNPATH, or other environment variables in the Workspace Properties dialog box to completely replicate the execution environment. To open the Workspace Properties dialog box, choose **Workspace | Properties** from the **File** menu. Programs running under CodeWatch also inherit any previously set environment variables.

### Exercise 1—Start CodeWatch

To start CodeWatch:

1. Start Microsoft Windows.
2. Click the Windows **Start** menu, point to **Programs**, point to **RMCOBOL** and then **Development v12**, and then click **CodeWatch**. On Windows 10, click the Windows **Start** menu, scroll to and select **RM Development v12**, and then click **CodeWatch**.
3. CodeWatch starts and displays the first page of the Workspace wizard. By default, the **Create New Workspace** option is selected.



To navigate quickly through the other pages of the Workspace wizard, press **Enter**. The options on all these pages of the Workspace wizard are described in more detail in [CodeWatch Workspace Wizard](#) (on page 35).

Once you save a workspace, the next time the wizard opens you will see a list of recently used configurations in the **Recent Workspaces** list. This allows you to select the **Use a Saved Workspace** option and then choose a saved workspace (with preset options) from the list, which can be used as a template, if desired, for a new workspace.

**Tip** To enter an RM/COBOL Runtime Command line quickly, click the **Command** button in the Workspace wizard. In the Set/Edit Command Line dialog box, type the command line in the edit box. Be sure to start with the name of the main program or program file. When you click **OK**, the fields in the Workspace wizard will be filled in from the command line that you entered. While the wizard is open, you can also press the **Command** button at any time to view or directly change the command line.

See the next section and exercise for steps on how to configure a new workspace.

---

## Configuring a Workspace

Once you have started CodeWatch, you will need to tell CodeWatch how to run your program. At a minimum, CodeWatch needs to know the name of your main program or program file and the directory where you wish to run the program. However, most RM/COBOL applications consist of many separately compiled programs and have additional requirements, which include:

- COBOL program files and libraries
- Configuration files
- Filename synonyms and other environment variables
- Runtime Command line options
- Source files
- Compilation options
- Directories to search for program files and source files

CodeWatch combines all of this information, and more, into a workspace. For convenience, a workspace can be saved to a file for later retrieval.

There are several ways to create a new workspace:

- Start CodeWatch using any of the methods discussed in [Starting CodeWatch](#) (on page 17) without any RM/COBOL Runtime Command line options specified.
- Start CodeWatch with a **runcobol** command line. See [Starting CodeWatch with a Runtime Command Line](#) (on page 19).
- If CodeWatch is already running, from the File menu, choose **Workspace | New**.

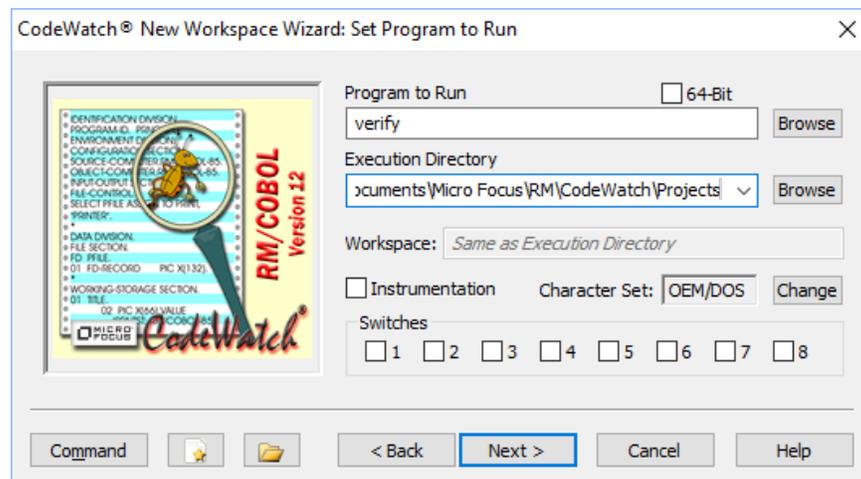
The Workspace wizard opens and then leads you through the process of creating the workspace. The name of the program or program file you wish to run is all that is required (although the working directory may also need to be specified).

**Note** You can bypass the Workspace wizard by clicking on the Command button in the wizard and entering a command line. To start CodeWatch without starting the wizard, specify /n on the command line. For more details, see [Starting CodeWatch from a Command Prompt](#) (on page 18).

## Exercise 2—Configure a Workspace with the Workspace Wizard

In this exercise, you will configure a workspace using the Workspace wizard.

1. Continuing from [Exercise 1](#) (on page 19), on the Introduction (or first) page of the wizard, click **Next** or press **Enter**. The **Set Program to Run** page is displayed.



2. Type **verify** in the **Program to Run** text box.
3. If necessary, click the **Execution Directory** dropdown and select

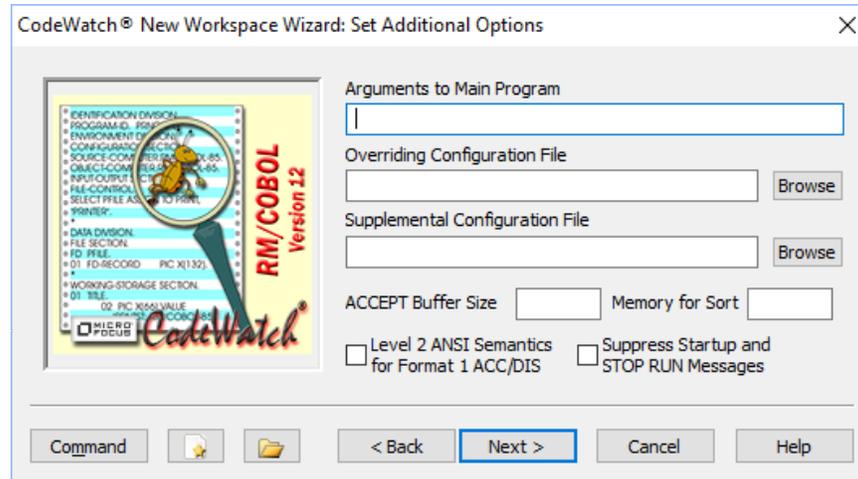
```
C:\Users\Public\Documents\Micro Focus\RM\CodeWatch\Projects
```

Then click the **Browse** button

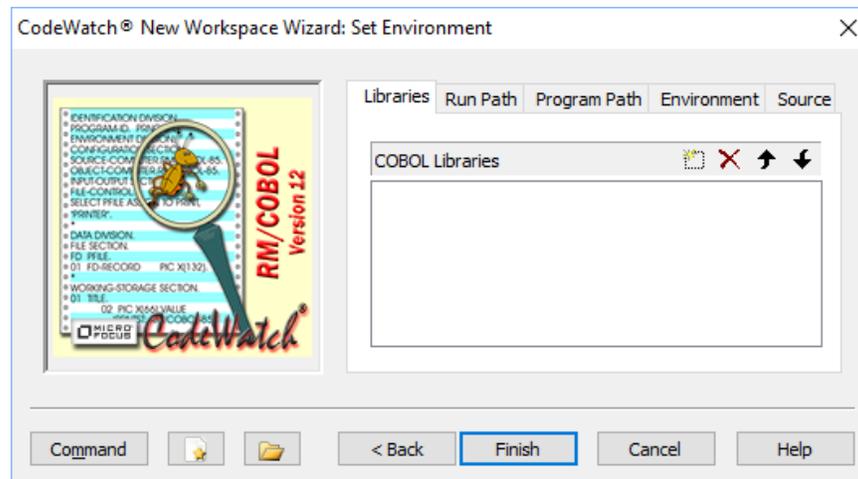
4. If you started CodeWatch from the Windows Start menu or Windows Explorer, the current directory will be set correctly. If not, click the **Browse** button or enter the name of the directory where the CodeWatch samples are installed in the **Working Directory**

field. Typically, this directory is named  
 C:\Users\Public\Documents\Micro Focus\RM\CodeWatch\Projects\Verify.

5. To use the 64-bit RM/COBOL runtime system, click the **64-Bit** checkbox. Note that this checkbox is only enabled if both 32-bit and 64-bit runtime systems are installed. Otherwise, the checkbox is disabled and is checked if a 64-bit runtime system is detected or is cleared if a 32-bit runtime system is detected.
6. Click **Next** or press **Enter** to move to the **Set Additional Options** page. This page allows you to enter program arguments or select configuration files. In this exercise, it is not necessary to change any of these settings.



7. Click **Next** or press **Enter** to move to the **Set Environment** page.



On this page, you can do one or more of the following:

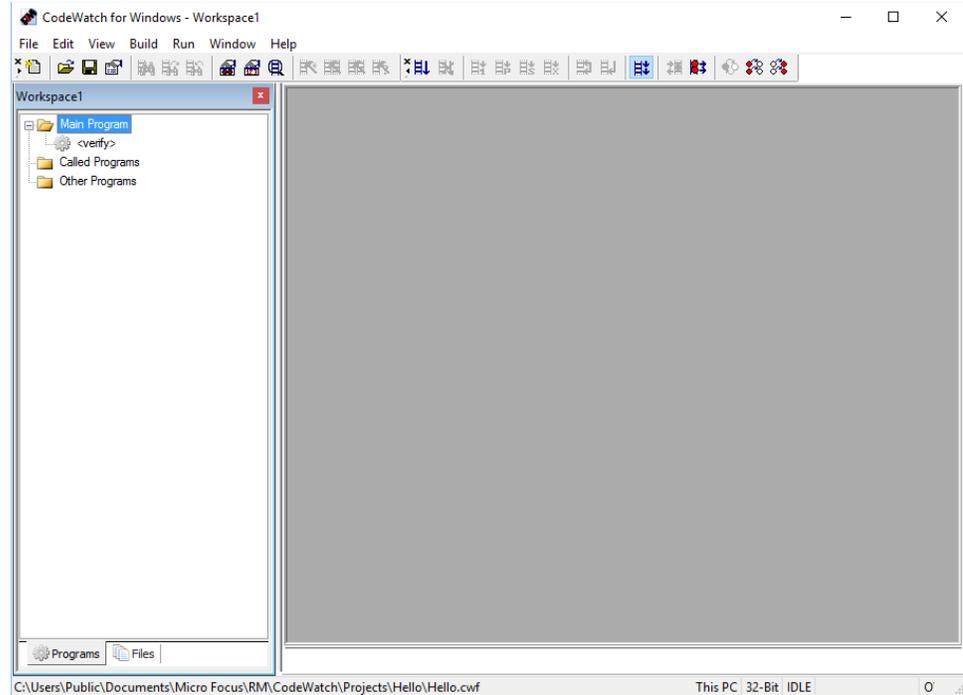
New Workspace Wizard: Set Environment	
Click this tab	To specify the
<b>Libraries</b>	COBOL program and dynamic-link libraries (DLLs). (This is equivalent to using the L= Runtime Command Option.)

New Workspace Wizard: Set Environment	
Click this tab	To specify the
<b>Run Path</b>	Directories to place into the RUNPATH environment variable. These directories are searched in the specified order to locate RM/COBOL programs (.cob) and application data files.
<b>Program</b>	Contents of the PATH environment variable. These are searched in the specified order to locate applications (.exe, .com, and .bat) and application extensions (.dll).
<b>Environment</b>	Additional options to place into the Windows environment, such as filename aliases (synonyms). This feature is provided for backward compatibility. Micro Focus recommends that aliases be placed in the Windows registry file using the RM/COBOL configuration utility ( <b>rmconfig.exe</b> ). For more information, refer to the “Windows Registry” and “Setting Synonym Properties” topics in Chapter 3: <i>Installation and System Considerations for Windows</i> of the <i>RM/COBOL User’s Guide</i> .
<b>Source</b>	List of directories to be searched for source files. This list is used if a source file cannot be located using the information embedded in the program file or library file. These directories are also placed in the RMPATH environment variable when compiling files and are used by the compiler to locate source and copy files.

### Tips

- Remember that you can press **Command** at any time from within the Workspace wizard and manually enter options using the Runtime Command line syntax. You can also view the command line that will be used to run the program by clicking this button.
  - Once the wizard is closed, you can change runtime options at any time. From the **File** menu, choose **Workspace | Properties** to open the [Workspace Properties dialog box](#) (see page 80).
8. For this example, do not change any settings on the Set Environment page. Simply click **Finish** or press **Enter**.

Configuration of a workspace is now complete. The Workspace window, displayed on the left side of the frame of the CodeWatch main window, shows the contents of your CodeWatch workspace.



The only item in the Main Program folder is the name of the main sample program, `<verify>`. The angle brackets indicate that possible ambiguities related to the program name have not been resolved; that is, `verify` may refer either to a standalone program named `VERIFY.COB` or to a program named `verify` in a COBOL library that has not yet been configured. CodeWatch cannot resolve these ambiguities until execution begins, as you may not yet have completed configuring the workspace.

**Note** At the bottom of the Workspace window, the **Programs** and **Files** tabs allow you to choose between two views of the workspace. The **Programs** tab lists the programs in the workspace. The **Files** tab lists all of the files in the workspace. Currently, there are no files in the Files tab because the program file that contains program `verify` is not discovered until the workspace is run for the first time. For a more complete description of the CodeWatch windows and user interface, see [Chapter 3: CodeWatch Interface](#) (on page 35).

---

## Viewing and Editing Program Source

You can view or edit the source for a program by double-clicking the name of the program, or the name of the program file in any window where the name appears. You can also directly open the source for a program file, a library file, or a source file listed in the Workspace window.

If the source file does not appear, it may be due to one of the following reasons:

- The program or library was not compiled with the Full Debug Information (Y=2 or Y=3) Compile Command line option (see Chapter 6: *Compiling in the RM/COBOL User's Guide*).
- The source file is not in the same location as when the file was originally compiled, and it could not be located using the search sequence described in the following exercise.

- There are program name ambiguities that have not yet been resolved. In this case, the name is shown in angle brackets.

### Exercise 3—Locate Source Files

To locate the source file for a COBOL program, CodeWatch uses the following search sequence:

1. First, the absolute path of the source file that was used to produce the program file is searched. Note that the file might have been compiled on a different computer, and, therefore, may not be found on this computer using this technique.
2. Then, the directory that contains the program file is searched, even if the source file was not compiled in this directory.
3. Next, the current working directory is searched.
4. Finally, each directory in the workspace **Source Path** list is searched. To view and add directories to this list, take the following steps:



- a. From the **File** menu, click **Workspace | Properties** or click the **Workspace Properties** toolbar button to display the Workspace Properties dialog box.
- b. In the Workspace Properties dialog box, click the **Environment** tab.
- c. On the Environment page, click the **Source** tab.
- d. Click the **New (Insert)** button (or double-click on the first blank line in the list box) to add a new line.
- e. Type the name of the directory that contains your source file and press **Enter**.  
If necessary, click the ellipsis button at the end of the line to open a Select Directory dialog box where you can search for and select the directory.
- f. To insert additional directories, click the **Move Down** button (or press **Alt+Down Arrow**) and then click in the list box.



- Note** You can reorder the list of directories by making a selection and pressing the **Move Up** and **Move Down** buttons (or the shortcut keys, **Alt+Up Arrow** and **Alt+Down Arrow**, respectively) or by dragging the items in the list with the mouse.
- g. Click **OK** to close the Workspace Properties dialog box.

**WARNING** CodeWatch will load the first occurrence of a given filename that is found when following this search sequence. It is possible that an incorrect file will be displayed if another file of the same name is found first.

---

## Running the Program

Once the workspace has been created, you can run the program. Program files may be produced either by CodeWatch using the **Build** menu or outside CodeWatch by the standalone RM/COBOL compiler. Note that, in order for CodeWatch to display the source for a program, the RM/COBOL compiler /Y2 or /Y3 options must be used.

For the example workspace, it is not necessary to configure any COBOL libraries since **verify** is a standalone program and is ready to run.

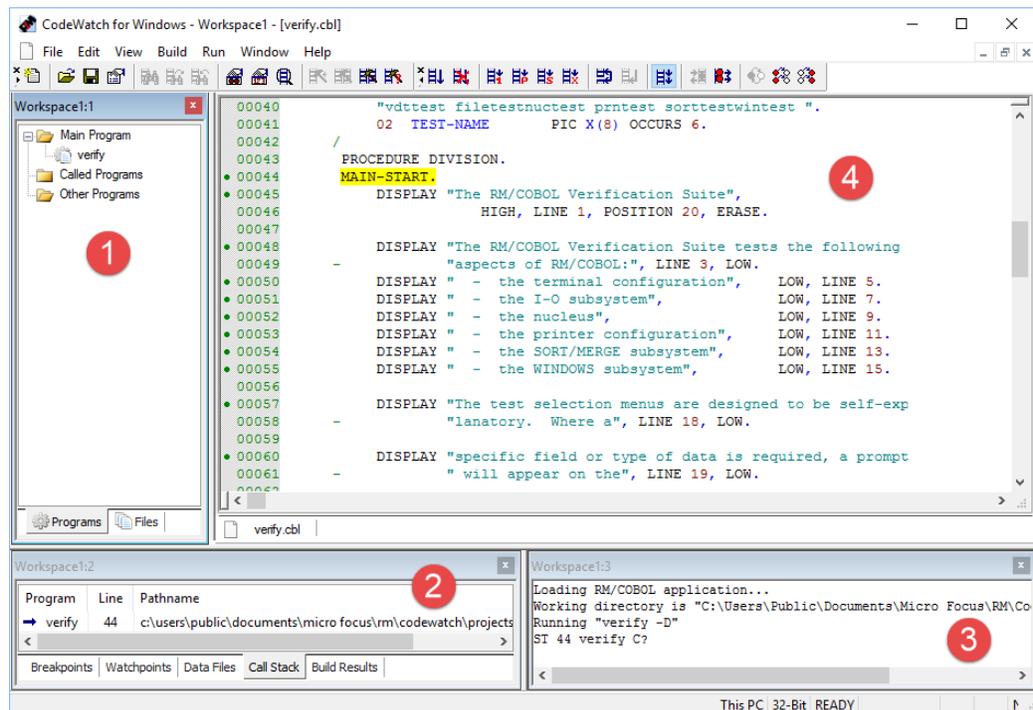
As you run your program, you can also perform the following tasks:

- [Animating the program](#) (see page 28)
- [Controlling the flow of execution](#) (see page 28)
- [Setting breakpoints](#) (see page 29)
- [Setting watchpoints](#) (see page 30)
- [Inspecting and modifying data](#) (see page 31)
- [Stopping the program](#) (see page 32)

## Exercise 4—Run the Program

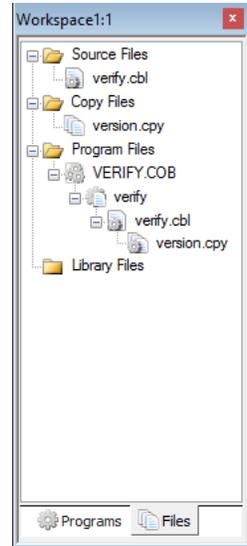
1. To start the program, do one of the following:
  - Press **F5** (the easiest way to run the program).
  - From the **Run** menu, choose **Start**.
  - Click the **Run** toolbar button.

CodeWatch will load the RM/COBOL runtime system using the workspace options you specified in the previous exercise. The runtime, in turn, locates and loads program **verify**. Once the RM/COBOL runtime has been started successfully, in addition to the Workspace window, three new windows will be visible, as shown in the following figure.



Counterclockwise from the upper left, the following windows are displayed:

1. **Workspace window.** Note that the angle brackets around program <verify> have been removed and that the name **verify** is shown in lowercase because the program ID was written quoted in lowercase when the program was compiled. A new item, **VERIFY.COB**, has been added to the Program Files folder listed on the Files tab. The **verify** program was located in the file **VERIFY.COB** in the current directory.



Click the **Files** tab, then [+], if necessary, to expand the item. You will see a list of objects in the COBOL library. In this case, **VERIFY.COB** contains only one program named **verify**. Because the program was compiled with the Full Debug Information (Y=2 or Y=3) Compile Command line option, an icon for the source file is also displayed.

2. **Data window.** This window contains a set of five tabs: Breakpoints, Watchpoints, Data Files, Call Stack, and Build Results. Click a tab to view the available options and settings for each.
3. **Log window.** This window allows you to interact with the RM/COBOL symbolic Debugger using a command line. You can click in this window and type RM/COBOL Debug commands while the program is running. This window is also used by the compiler to display progress messages during a build operation.
4. **Source Code/Edit Window.** This window displays the source for the program that you are currently running. While you are debugging, the statement that is about to be executed is highlighted. To display multiple Source Code/Edit windows simultaneously, simply click the **Restore** button in the upper-right corner of the title bar. You may have as many Source Code/Edit windows visible simultaneously as you like, or you can maximize the window and see as much of a single source file at one time as possible. To maximize the window, click the **Maximize** button or double-click the title in the Source Code/Edit window.

You can also split a Source Code/Edit window either horizontally or vertically. From the **Window** menu, choose **Split**. Alternatively, you can drag the splitter bar next to the vertical or horizontal scroll bar to split the window. Splitting the window allows you to see non-adjacent sections in a source file.

You can open the source for any program for which source is available by double-clicking the source file, program file, or library file icon in the Files tab of the Workspace window, or the program in the Programs tab of the Workspace window.

**Note** These windows are described in more detail in [Chapter 3: CodeWatch Interface](#) (on page 35) and [Chapter 4: Source Code Editor](#) (on page 97).

2. Once the program has been started, you can continue to run the program in any of the following ways:

Running the program		
To run	Choose	Or click
To the next breakpoint, watchpoint, end of program, or until you choose to break the program	<b>Resume</b> on the Run menu or press F5	Run
One statement	<b>Step Statement</b> on the Run menu or press F8	Step Statement

Running the program		
To run	Choose	Or click
One paragraph or to the end of the current paragraph	<b>Step Paragraph</b> on the Run menu or press Shift+F8	 Step Paragraph
One section or to the end of the current section	<b>Step Section</b> on the Run menu or press Ctrl+F8	 Step Section
To the end of this program	<b>Step Program</b> on the Run menu or press Alt+F8	 Step Program
To run to the next program (Note that this is either the next program called from within this program or the program that called the current program.)	<b>To Program Change</b> from the Run menu or press Alt+F5	 Run To Program Change
To run to the line containing the cursor in the Source Code/Edit window	<b>To Cursor</b> on the Run menu or press Shift+F5	 Run to Cursor

## Animating the Program

The animation feature determines whether CodeWatch will trace program execution while running. When enabled, CodeWatch opens the source code for each program in a Source Code/Edit window and highlights each line as it is executed. While this is beneficial for visually observing the flow of control within your program, there is a significant performance penalty for running the program with animation enabled.

**Note** You can turn animation on and off at any time during program execution.

### Exercise 5—Animate the Program



- Continuing from [Exercise 4](#) (on page 26), click **Animate** from the **Run** menu to toggle animation on or click the **Animate** toolbar button.

**Note** To indicate that this option is selected, a check mark appears next to the menu command.

## Controlling the Flow of Execution

Once the program begins to execute, CodeWatch will run the program until one of the following events occurs:

- A breakpoint is encountered during execution.
- A watchpoint is encountered during execution.
- The operation has completed, if you are stepping across lines, paragraphs, or sections.
- You tell the program to stop running by choosing **Break** from the [Run menu](#) (see page 77).
- The program terminates for one of the following reasons:

- You choose **Stop** from the **Run** menu.
- The end of the RM/COBOL program is reached.
- An error occurs in the RM/COBOL program.
- You terminate CodeWatch or close the CodeWatch main window.

If the program halts for any of the above reasons except for the last, you can resume program execution with the **Resume** command from the **Run** menu or any of the other commands on the **Run** menu. Alternatively, you can choose **Stop** from the **Run** menu to terminate the program.

## Setting Breakpoints

A breakpoint causes program execution to pause, which gives you the opportunity to look around at the program, examine variables, or set additional breakpoints.

Breakpoints are part of the workspace and are, therefore, saved and loaded with the workspace. Any breakpoints that are set when you save a workspace will be set the next time you open the workspace.

Breakpoints are set by program name, line number, and statement number. If you have two programs in your workspace with identical names, and both programs contain the line number on which the breakpoint was set, execution will stop at both lines.

If a line contains a breakpoint, the Source Code Editor displays a red circle in front of the line number. The breakpoint is also reported in the [Breakpoints tab](#) (see page 52) of the Data window. If the breakpoint is enabled, the red circle is solid; if the breakpoint is disabled, the red circle is hollow.

### Exercise 6—Set a Breakpoint

1. Continuing from [Exercise 5](#) (on page 28), scroll down through the Source Code/Edit window until DISPLAY-MAIN-MENU is visible.
2. To set a breakpoint, click anywhere in the line and press the **F9** key.

A red circle appears to the left of the line number and the breakpoint appears in the Breakpoints tab.

**Note** In addition to pressing **F9**, you can set breakpoints by using any of the following methods:



- Click on the line in the [Source Code/Edit window](#) (see page 97) and choose **Breakpoint | Set** from the **Run** menu, or click the **Toggle Breakpoint** toolbar button.



- Right-click anywhere in the line in the Source Code/Edit window and choose **Set Breakpoint** from the pop-up menu. Note that you can also remove breakpoints, or temporarily disable and later enable breakpoints by using this menu.
- Press **Ctrl+B** or click the **Edit Breakpoints** toolbar button to display the Breakpoints & Watchpoints dialog box and interactively set, remove, or modify breakpoints using the [Code Breakpoints tab](#) (see page 88).
- Display the [Breakpoints tab](#) (see page 52) in the Data window by choosing **Breakpoints Tab** from the **View** menu or pressing **Alt+1** to display this tab in the window, if not visible. Then, press **Ins** to insert a new breakpoint, **Del** to delete a breakpoint, click the

icon (solid red circle) to enable or disable a breakpoint, or press **F2** to examine or change a breakpoint's properties in the Code Breakpoints page of the Breakpoints & Watchpoints dialog box.



- Alternatively, you can cause a break at any time by choosing **Break** from the **Run** menu, pressing **F5**, or by clicking the **Run** toolbar button again while the program is running.

## Setting Watchpoints

A watchpoint allows you to examine or modify the contents of your COBOL data items. Watchpoints are displayed in the [Watchpoints tab](#) (see page 53) in the Data window and are updated whenever a program break occurs. You can also request that a program break occur when there is a change in the value of the data item being watched.

Watchpoints are part of the workspace and are, therefore, saved and loaded with the workspace. Any watchpoints that are set when you save a workspace will be set the next time you open the workspace.

### Exercise 7—Set a Watchpoint

1. Continuing from [Exercise 6](#) (on page 29), scroll down through the Source Code/Edit window until ACCEPT-MAIN-MENU is visible.
2. Double-click on FUNCTION-NUMBER on the next line to highlight the word.
3. Right-click and choose **Watch Data Item** from the pop-up menu to set the watchpoint.

The Data window switches to the Watchpoints tab. FUNCTION-NUMBER and its location, data type, and present value appear in the window.

4. Click on FUNCTION-NUMBER in the Watchpoints tab and press the **Spacebar** so that execution will stop when the value in FUNCTION-NUMBER changes.

Alternatively, single click on the solid red circle to left of the data item name. It will turn into a hollow circle.



5. Choose **Resume** from the **Run** menu, click the **Run** toolbar button, or press **F5** to run the program.

Since animation is enabled, CodeWatch displays each line as it is executed. When the ACCEPT RETURN-KEY statement is reached, the RM/COBOL program output window will appear.

6. Press **Enter** and CodeWatch will stop at the watchpoint that you set on the ACCEPT-MAIN-MENU line.
7. Press **F5** to continue execution.

The runtime will request a test number.

8. Type **3** and press **Enter**.

Execution will stop again because FUNCTION-NUMBER has changed.



9. Choose **Step Statement** from the **Run** menu, click the **Step Statement** toolbar button, or press **F8** to step over the current statement.
10. Press **F8** again and you should see INIT-PARA in program NUCTEST.

The program **NUCTEST** has now been added to the Called Programs folder in the Workspace window, and the program file **NUCTEST.COB** has been added to the list of program files in the Workspace window.

**Note** In addition to the method described in the preceding exercise, you can set watchpoints by using any of the following methods:

- Highlight and then right-click a data item name in any Source Code/Edit window and choose **Watch Data Item** or press **Alt+F9** from the pop-up context menu.
- Double-click a data item name in any Source Code/Edit window to display the Inspect/Modify dialog box. After examining and possibly modifying the value, you can click the **Watch** button in that dialog box to set a watchpoint (see the following section).
- Press **Ctrl+W** or click the **Edit Watchpoints** toolbar button to display the Breakpoints & Watchpoints dialog box and interactively add, remove, modify, or enable or disable a program break when the value of the watchpoint changes using the [Data Watchpoints page](#) (see page 89).
- Display the [Watchpoints tab](#) (see page 53) in the Data window by choosing **Watchpoints Tab** from the **View** menu or pressing **Alt+2** to display this tab in the window, if not visible. Then, press **Ins** to insert a new watchpoint, **Del** to delete a watchpoint, click the icon (solid red circle) to enable or disable a watchpoint, or press **F2** to change a watchpoint's properties in the Data Watchpoints page of the Breakpoints & Watchpoints dialog box.

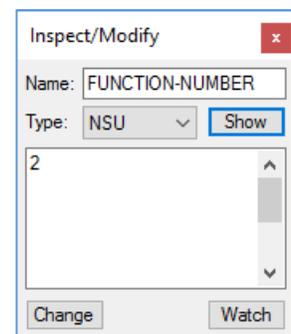


## Inspecting and Modifying Data

CodeWatch provides the Inspect/Modify dialog box, illustrated below, for inspecting and modifying data.

Any of the following actions cause the Inspect/Modify dialog box to appear:

- In a Source Code/Edit window, double-click the name of a data item, or select a data item and then double-click within the selection.
- In a Source Code/Edit window, select a data item and then right-click within the selection and choose **View Data Item** from the pop-up menu, or press **Ctrl+I** or **Shift+F9**.
- From the **View** menu, choose **Data Item**.



### Exercise 8—Select a Data Item

If you have not yet selected a data item, take the following steps:

1. Type an identifier in the **Name** text box.

**Note** For a description of allowed expressions, see “Data Address Development” in Chapter 9: *Debugging of the RM/COBOL User's Guide*.

2. In the **Type** list box (this is an optional field), select or enter a data type. For a description of allowed data types, see “Data Types” under the “General Debug Concepts” section in Chapter 9: *Debugging* of the *RM/COBOL User’s Guide*.
3. Click **Show** or press **Enter**.

The current value of the identifier is then displayed in the **Show** text box using either the selected type or the actual type of the identifier. If the identifier cannot be displayed, an error message is displayed in red. Note that the program must be running and the identifier must be “in scope” before a value can be displayed. See the description of the Show option in the [Inspect/Modify dialog box](#) (on page 66).

To change the value of a data item:

- In the **Show** text box, type the new value and click **Change**.

Be sure that the entered value is compatible with the displayed data type. You can also change the type before clicking **Change** to set the type using a different encoding style (for example, hexadecimal).

**Note** The **Change** button is not enabled if the data item being inspected is a constant-name. Also note that you can select text with the cursor keys in either the **Name** or the **Show** text box, and right-click to pop up a menu that allows text to be copied to the Windows Clipboard or pasted from the Clipboard. You can use this technique to copy the contents of one data item into another data item.

To set a watchpoint for the data item:

- Click **Watch**.

This is a convenient way to set a watchpoint on an item other than a simple variable; for example, a group item or an array element.

## Stopping the Program

There are several methods you can use to stop your program.

### Exercise 9—Stop the Program



1. From the **Run** menu, choose **Stop** or click the **Stop Run** toolbar button.

This action sends a stop request to the RM/COBOL runtime system. The runtime will display a termination message in the RM/COBOL program output window. If this is the case, single-step or choose **Stop** again to dismiss the output window.

**Note** If you have set the Persistent runtime property, you will have to close the RM/COBOL program output window manually. See Chapter 3: *Installation and System Considerations for Microsoft Windows* of the *RM/COBOL User’s Guide*.

2. Switch to the RM/COBOL program output window and close the window.

**Note** Closing either the Workspace window or the CodeWatch main window will also stop the program.

## Changing the Workspace

It is often desirable to change properties or runtime options, or add libraries to a workspace after it has already been created.

To change the workspace:

1. From the File menu, choose **Workspace | Properties**.

The [Workspace Properties dialog box](#) (see page 80) appears. This tabbed dialog box contains the same options that are displayed on the pages of the Workspace wizard.

2. In the Workspace Properties dialog box, select the page for the property that you want to change.

---

## Saving the Workspace

To save the workspace for a subsequent debugging session:

- From the **File** menu, choose **Workspace | Save** or press **Ctrl+Shift+S**.

CodeWatch will prompt you for a name if the workspace has not previously been saved.

To save the workspace under a new name:

1. From the File menu, choose **Workspace | Save As** or press **Ctrl+Shift+A**.
2. At the prompt, enter a new name for the workspace.
3. Click **OK**.

The workspace is saved with the new name.

The workspace is saved in a file with an extension of **.cwf** (**C**ode**W**atch **W**orkspace **F**ile).

This file is an ASCII text file that can be examined and modified with any text editor. If you have a COBOL application with a large number of libraries, it may, in some cases, be easier to create a workspace, close CodeWatch, and then add the libraries to the workspace file using a text editor. This practice, however, is not recommended.

**Note** When you close CodeWatch or the Workspace window, you will be prompted to save the workspace if any changes have been made.



# Chapter 3: CodeWatch Interface

This chapter provides detailed descriptions of the Workspace wizard, and the various windows, dialog boxes, menus, and toolbars found in CodeWatch.

**Note** The information provided in this user's guide is for traditional CodeWatch for Windows source code debugging. For BIS-based debugging on an IIS web server, see the *CodeWatch for Xcentrivity Business Information Server User's Guide Supplement (Version 12)*.

---

## CodeWatch Workspace Wizard

The Workspace wizard helps guide you through the steps to create and configure a new workspace. A workspace contains the name of your main program or program file and the directory where you wish to run the program. A workspace can also include information about the following:

- COBOL program files and libraries
- Configuration files
- Filename synonyms and other environment variables
- Runtime Command line options
- Source files
- Compilation options
- Directories to search for program files and source files.

A workspace is displayed in a Workspace window in the [CodeWatch main window](#) (see page 43). Workspace files have a **.cwf** filename extension.

**Note** CodeWatch does not require that you configure a workspace before you open or create a source file. It will create a default workspace, as required.

The CodeWatch Workspace wizard is displayed when you start CodeWatch without a command line parameter. If CodeWatch is running and no workspace is open, you can display the wizard by choosing the **New | Workspace** command from the **File** menu. For more information, see [Starting CodeWatch](#) (on page 17).

If a workspace is already open for the session, choose **Workspace | New** from the **File** menu to open the wizard. However, since only one workspace may be open at a time, the workspace must be closed before you can open or create a new one. CodeWatch will prompt

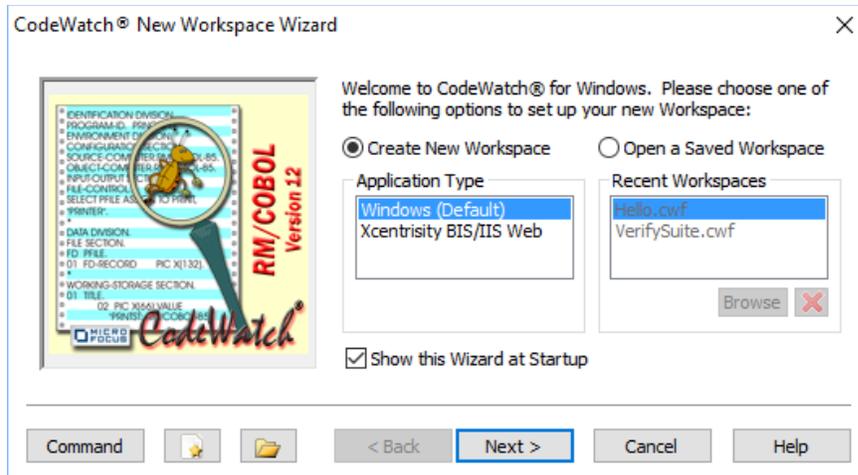
you to save the workspace if the workspace has not previously been saved, or has been changed since the last save. You may also save and close the workspace by choosing the **Workspace | Save** command from the **File** menu. To save and rename the workspace, choose **Workspace | Save As** from the **File** menu.

The Workspace wizard consists of four pages: Introduction (New), Set Program to Run, Set Additional Options, and Set Environment. Several buttons appear along the bottom of the Workspace wizard no matter which page is selected and displayed. They perform the same function or action on all the pages of the wizard and are described below.

Workspace Wizard	
Use this button	To
<b>Command</b>	Open the Select/Edit Command Line dialog box where you can view or edit the RM/COBOL Runtime Command line options. This option is for advanced users.
	Cancel the Workspace wizard and create a new source file. A workspace can automatically be created when the file is saved.
	<p>Cancel the Workspace wizard and display an Open a COBOL File or a Workspace dialog box where you can open an existing workspace file or any text file into the Source Code/Edit window. Select the Add to Workspace check box to create a workspace automatically for one of the following files:</p> <ul style="list-style-type: none"> <li>• COBOL Source File</li> <li>• COBOL Copy File</li> <li>• COBOL Program File</li> <li>• COBOL Program Library File</li> </ul> <p>You can then use either the Open or Insert Items command from the File menu to add more files to the workspace. In all the cases above, the working directory is set to the directory containing the file.</p> <p>To <a href="#">configure the workspace</a> (see page 80), choose <b>Workspace   Properties</b> from the File menu. See also <a href="#">Configuring a Workspace</a> (on page 20).</p>
<b>Back</b>	Display the previous page of the wizard. (Note that this button is grayed out on the first page of the wizard.)
<b>Next</b>	Display the next page of the wizard. (Note that this button does not appear on the last page of the wizard.)
<b>Finish</b>	Create the workspace using the settings from this and the preceding pages. (Note that this button appears only on the last page of the wizard.)
<b>Cancel</b>	Cancel the wizard.
<b>Help</b>	Display help for this page of the wizard.

## Introduction (New) Page

This page introduces the Workspace wizard and allows you to select the type of application that you will be debugging. You can also open a saved workspace that will be used as a template for the new workspace.

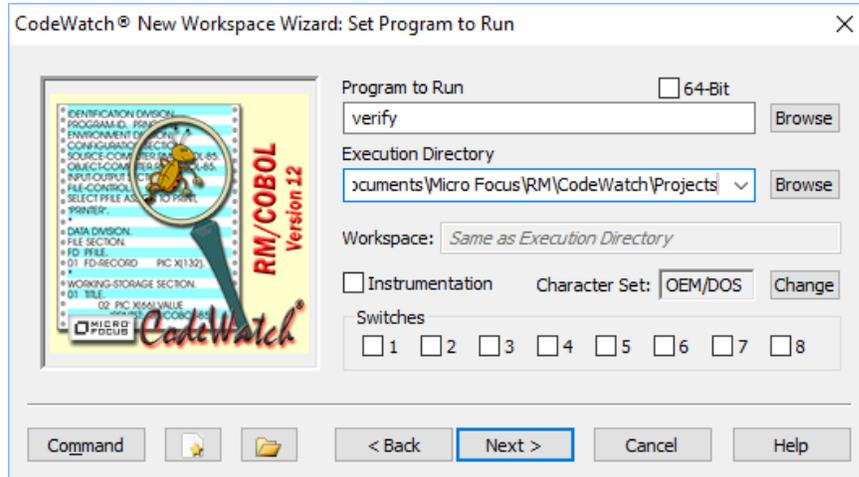


New Workspace Wizard	
Use this option	To
<b>Create New Workspace</b>	<p>Create a new workspace without initializing any of the options on the subsequent pages of the Workspace wizard.</p> <p>In the <b>Application Type</b> list box, select the type of application you will be debugging.</p> <p><b>Note</b> If you select the Xcentrinity BIS/IIS Web option, see the <i>CodeWatch for Xcentrinity Business Information Server User's Guide Supplement (Version 12)</i> for more information.</p>
<b>Open a Saved Workspace</b>	<p>Choose a saved workspace from the list of your most recently used workspaces. The settings of the saved workspace will be used to create the new workspace.</p> <p><b>Note</b> This option simply reads the preset options from the previous workspace. It does not open the workspace.</p> <p>In the <b>Recent Workspaces</b> list box, select a workspace file (.cwf) from the list of recently used workspaces on the selected computer.</p> <ul style="list-style-type: none"> <li>Use the <b>Browse</b> button to open the Select a Workspace File dialog box and search for saved workspaces that are not in the recently used list.</li> <li>Click  to remove the selected workspace from the list of recently used workspaces. This action does not delete the actual workspace file from your disk, but is useful for removing workspaces that no longer exist from this list.</li> </ul>
<b>Show this Wizard at Startup</b>	Determine whether the Workspace wizard is displayed when CodeWatch is started. This option is selected by default.

For a description of the buttons that appear along the bottom of this page, see the [CodeWatch Workspace wizard](#) (on page 35).

## Set Program to Run Page

The Set Program to Run page of the Workspace wizard, illustrated below, allows you to select a program to run, choose a character set for the workspace, and to define the execution environment for the program.



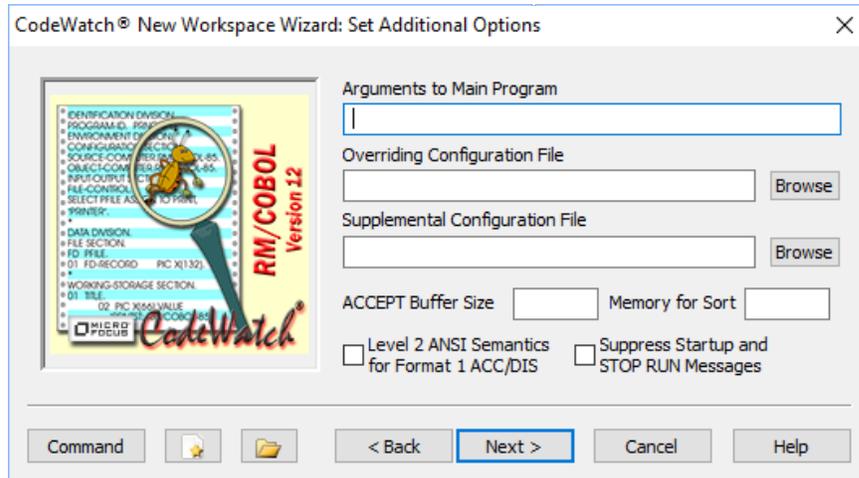
New Workspace Wizard: Set Program to Run	
Use this option	To
<b>Program to Run</b>	<p>Type or select the name of the COBOL program that CodeWatch is to run. This can be one of the following:</p> <ul style="list-style-type: none"> <li>The filename of a COBOL program file (for example, VERIFY.COB).</li> <li>The name of a COBOL program to be found in a COBOL library file (for example, VERIFY).</li> </ul> <p>CodeWatch uses the RM/COBOL runtime to locate programs (see the “Subprogram Loading” topic in Chapter 8: <i>RM/COBOL Features of the RM/COBOL User’s Guide</i>, for a complete description of the methods). For example, if VERIFY is specified as the program to run, and VERIFY is not found in any configured library, then the RM/COBOL runtime system will optionally search for a synonym for VERIFY in the environment, and will then search for a file named VERIFY.COB that contains exactly one separately compiled COBOL program in the current directory, the PATH, and the RUNPATH. CodeWatch does not attempt to resolve an ambiguous main program name until execution begins.</p> <p>This box may be left blank. If a main program is not specified, you will be asked for the program name when you run the workspace.</p> <p>Click the <b>Browse</b> button to open the Select a Program File dialog box where you can search for and select a COBOL program file (.cob).</p>
<b>64-Bit</b>	<p>When checked, the 64-bit runtime system will be used. The checkbox is disabled and unchecked if only a 32-bit runtime system is installed, and is checked and disabled if only a 64-bit runtime system is installed.</p>

New Workspace Wizard: Set Program to Run	
Use this option	To
<b>Working Directory</b>	<p>Select the current directory that is used when the COBOL program begins execution. Click the down arrow in the list box to choose from a list of recently used working directories.</p> <p>Click the <b>Browse</b> button to open a Select a Directory dialog box where you can search for and select a working directory.</p>
<b>Workspace</b>	<p>This read-only field is the root location of the workspace. For new workspaces, this is initially the same location as the Working Directory, but is set to the directory that contains the workspace file when the workspace is loaded or saved.</p> <p>Whenever possible, the files in the workspace are processed using this directory as a base directory. By saving the workspace into a directory that contains all of your source files, COPY files, and object files, those filenames can be stored as relative pathnames. This makes workspace files portable between machines, and makes it possible to check workspace <b>.cwf</b> files into version control systems.</p>
<b>Instrumentation</b>	<p>When selected, enable instrumentation. This option is equivalent to the /I Runtime Command Option.</p>
<b>Character Set</b>	<p>Show the character set for this workspace. The character set is used in the editor, during compilation, and during execution. Two character sets are allowed:</p> <ul style="list-style-type: none"> <li>• OEM/DOS</li> <li>• ANSI/Windows</li> </ul> <p>For a new workspace, the character set defaults to OEM/DOS after an installation of RM/COBOL. The RMSETNCS utility may be used to change the default character set for new workspaces to ANSI/Windows.</p> <p>After a workspace is created, the character set for that workspace remains the same unless explicitly changed by the user. The default character set for new workspaces can be overridden by specifying one of the /cs_ansi or /cs_oem command line options when starting CodeWatch. For more information, see <a href="#">System Installation</a> (on page 8).</p> <p>To change the character set, click the <b>Change</b> button (this option is described below).</p> <p><b>Note</b> For a detailed explanation of character sets within CodeWatch, see <a href="#">Character Sets</a> (on page 104). For further information about character sets and the RMSETNCS utility in RM/COBOL, see "Character Set Considerations for Windows" in Chapter 3: <i>Installation and System Considerations for Windows</i> of the <i>RM/COBOL User's Guide</i>.</p>
<b>Change</b>	<p>Open the <a href="#">Select Character Set dialog box</a> (see page 95) to change the character set assumed for the workspace. The dialog box also allows configuring the euro symbol for use in the CodeWatch editor.</p>
<b>Switches</b>	<p>Set or clear the current set of runtime switches. The eight switches are numbered consecutively from left to right. These options are equivalent to the /S Runtime Command Option.</p>

For a description of the buttons that appear along the bottom of this page, see the [CodeWatch Workspace wizard](#) (on page 35).

## Set Additional Options Page

The Set Additional Options page of the Workspace wizard, illustrated below, allows you to continue configuring the execution environment.



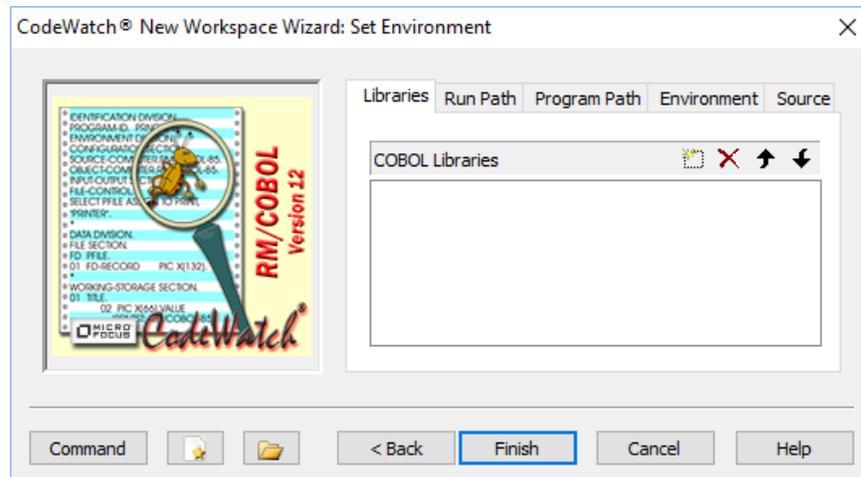
New Workspace Wizard: Set Additional Options	
Use this option	To
<b>Arguments to Main Program</b>	<p>Enter arguments to be passed to the main program. This option has the following format:</p> <p style="text-align: center;"><i>[delimiter]string[delimiter]</i></p> <p>The <i>delimiter</i> character may be either a single or double quote, must not appear within <i>string</i>, and is required only if <i>string</i> contains spaces or tabs. If you enter a string that contains spaces, but do not use a delimiter, CodeWatch will provide a delimiter or display an error if no delimiter can be chosen. This option is equivalent to the /A Runtime Command Option.</p>
<b>Overriding Configuration File</b>	<p>Select or enter the name of the primary runtime configuration file. This file completely overrides the default runtime configuration. This option is equivalent to the /C Runtime Command Option. If a relative path is specified, the path is interpreted as relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard.</p> <p>Click the <b>Browse</b> button to open the Select a Configuration File dialog box where you can search for and select a configuration file (.cfg).</p>
<b>Supplemental Configuration File</b>	<p>Select or enter the name of the supplemental runtime configuration file. The contents of this file supplement the default runtime configuration without replacing it. If an overriding configuration is specified, the contents of this file supplement that configuration. If no other configuration is specified, this file provides the complete configuration. This option is equivalent to the /X Runtime Command Option. If a relative path is specified, the path is interpreted as relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard.</p> <p>Click the <b>Browse</b> button to open the Select a Configuration File dialog box where you can search for and select a configuration file (.cfg).</p>

New Workspace Wizard: Set Additional Options	
Use this option	To
<b>ACCEPT Buffer Size</b>	Enter the maximum buffer size to be used by ACCEPT and DISPLAY statements. The maximum size is 65280 characters, and the default size is 264 characters. This option is equivalent to the /B Runtime Command Option.
<b>Memory for Sort</b>	Enter the amount of memory (in bytes) to be used for a sort operation. The default is 256000 bytes. Specify a larger value to improve the efficiency of sort operations. Decrease this value to allocate more memory for other data or called subprograms. If neither SORT nor MERGE operations will be used in this run unit, specifying 0 will free all reserved memory for other uses. This option is equivalent to the /T Runtime Command Option.
<b>Level 2 ANSI Semantics for Format 1 ACC/DIS</b>	When selected, indicate that Level 2 ANSI semantics are to be used for Format 1 ACCEPT and DISPLAY operations. The default is to use Level 1 ANSI semantics for these operations. This option is equivalent to the /M Runtime Command Option.
<b>Suppress Startup and STOP RUN Messages</b>	When selected, suppress the banner message and the STOP RUN message. This option is equivalent to the /K Runtime Command Option.

For a description of the buttons that appear along the bottom of this page, see the [CodeWatch Workspace wizard](#) (on page 35).

## Set Environment Page

The Set Environment page of the Workspace wizard, illustrated below, allows you to finish defining the execution environment for your RM/COBOL program. This page contains five tabs. Click on a tab to set the parameters for the options available.



New Workspace Wizard: Set Environment	
Use this option	To
<b>Libraries</b>	Specify a list of pathnames of COBOL and non-COBOL program libraries to be loaded. Libraries are loaded and searched in the specified order. The order can be changed by selecting a library and pressing the Move Up or Move Down buttons to move the name of the library within the list, or by dragging the name of the library to a new position with the mouse. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard. This option is equivalent to the /L Runtime Command Option.
<b>Run Path</b>	Specify a list of directories to be inserted before the RUNPATH environment variable. The RM/COBOL runtime will search these directories in the specified order for COBOL programs, libraries, non-COBOL libraries, and configuration files. These directories are inserted in front of any directories already in the RUNPATH environment variable. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard.
<b>Program Path</b>	Specify a list of directories to be inserted before the PATH environment variable. These directories are inserted in front of any directories already in the PATH environment variable. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard.
<b>Environment</b>	Specify a list of string pairs to be placed into the RM/COBOL program's environment. The syntax is:  <i>name=value</i>  Among other purposes, environment strings are used to specify synonyms for filenames. See the "Locating RM/COBOL Files on Windows" topic in Chapter 3: <i>Installation and System Considerations for Microsoft Windows</i> of the <i>RM/COBOL User's Guide</i> .
<b>Source</b>	Specify a list of directories to be searched for source files, in the event that a source file cannot be found at the location specified in the object file. These directories are searched in the specified order, and the first file with a matching name is used. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Set Program to Run page</a> (see page 38) of the wizard. For additional information, see <a href="#">Locating Source Files During Execution</a> (on page 97) and the "Viewing Source During Program Execution" topic in <a href="#">Using the Source Code/Edit Window</a> (on page 101).

The following buttons appear on each of the tabs of the Set Environment page and perform the same function on each tab.

New Workspace Wizard: Set Environment Buttons	
Button	Name and Description
	<b>New.</b> Inserts a new item.
	<b>Delete.</b> Removes the selected item.

New Workspace Wizard: Set Environment Buttons	
Button	Name and Description
	<b>Move Up</b> (Alt+Up Arrow) and <b>Move Down</b> (Alt+Down Arrow). Moves the selected item up or down in the list.
	<b>Ellipsis.</b> On tabs that specify files or directories, clicking this button displays a File Open dialog box in order to search for and select a file or directory.

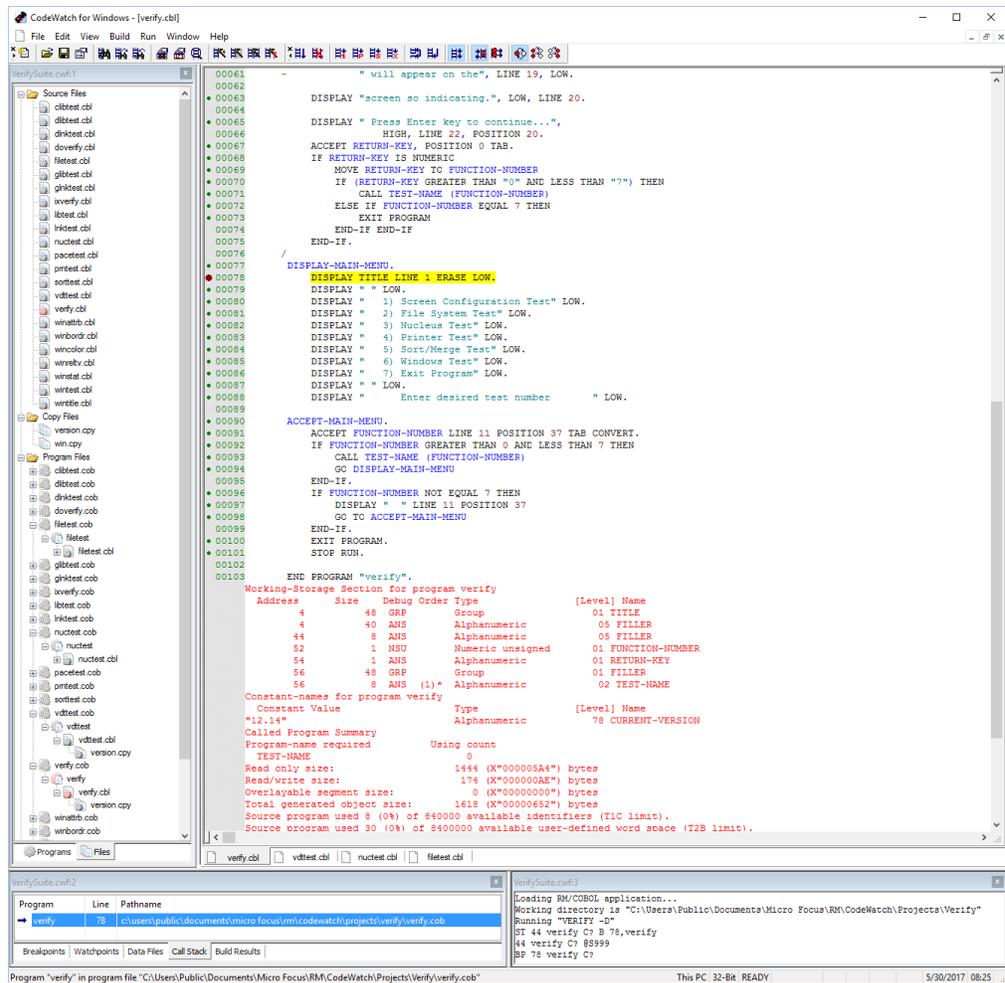
For a description of the buttons that appear along the bottom of this page, see the [CodeWatch Workspace wizard](#) (on page 35).

---

## CodeWatch Main Window

Once the Workspace wizard has created a workspace, you will see the CodeWatch main window, as shown in the following illustration. The main window consists of two general parts:

1. A window frame, consisting of a title bar, menu bar, and toolbar at the top of the window, and a status bar at the bottom. The frame surrounds the client area.
2. The client area. This is the area bounded by the frame and may contain zero or more Workspace windows.



The following briefly describes the elements of the window frame area of the CodeWatch main window.

CodeWatch Main Window	
Area	Description
<b>Title Bar</b>	Displays the text, “CodeWatch for Windows”, followed by the name of the current workspace and the file being displayed. You can move the CodeWatch main window by clicking the title bar and dragging with the mouse.
<b>Menu Bar</b>	Use drop-down menus on the menu bar to display CodeWatch commands and subcommands. The menus available depend on whether or not a program file has been opened. For more information, see <a href="#">Menus and Toolbars</a> (on page 68).

CodeWatch Main Window	
Area	Description
<b>Toolbar</b>	<p>Click buttons on the toolbar to choose menu commands quickly. The toolbar buttons available depend on several factors, including whether or not a program file has been opened and whether the program is running. For more details, see <a href="#">Menus and Toolbars</a> (on page 68).</p> <p>Note that an option on the View menu allows you to show or hide toolbars. You can move a toolbar at a different position in the CodeWatch main window by dragging the toolbar's gripper bar (the two etched lines at the left end of the toolbar) with the mouse. You can detach a toolbar from the main window and float it anywhere on the screen by holding down the Ctrl key while dragging the toolbar.</p>
<b>Status Bar</b>	<p>Displays current information about the targeted PC, the selected 32-bit or 64-bit runtime system, the execution state, the date, the time, and the keyboard state. The execution state field indicates the state of the program you are debugging and may display one of the following messages:</p> <ul style="list-style-type: none"> <li>• NO PROGRAM. No program has been configured into the workspace.</li> <li>• IDLE. The program is not currently running.</li> <li>• LOADING. The program is loading.</li> <li>• READY. The program has been loaded, is ready to run, and is awaiting a Debug Command.</li> <li>• RUNNING. The program is running.</li> <li>• ACCEPT. The program is awaiting keyboard input. If the running program is not visible, press Alt+Tab to switch to the program.</li> </ul>

The CodeWatch main window may contain zero or more Workspace windows. These windows are used to perform most of the actual work in CodeWatch. [Table 1](#) describes the characteristics of the three types of Workspace windows.

Table 1: Types of Workspace Windows

Types of Workspace Windows	
Window Type	Description
<b>Dockable Windows</b>	<p>Dockable windows have a small title bar, and are initially attached to the frame of the CodeWatch main window or to each other. Dockable windows can be:</p> <ul style="list-style-type: none"><li>• Docked to the inside of the frame of the main window.</li><li>• Floated outside of the frame of the main window anywhere on the screen.</li><li>• Converted to or from a document window. To convert a docked window into a document window or back again, choose Docked from the Window menu.</li><li>• To undock a docked window, choose Floating from the Window menu. You can then move the window around by dragging its title bar. To prevent the window from docking as you move it over the window frame, hold down the Ctrl key.</li><li>• Right-clicking within the dockable window provides access to a menu that allows you to toggle between Docked and Floating. In addition, right-clicking the title bar of a dockable window displays a menu with additional options that allow you to precisely control how and where the window docks.</li></ul> <p>For more information, see <a href="#">Dockable Windows</a> (on page 46).</p>
<b>Document Windows</b>	<p>Document windows have a large title bar and usually occupy a large part of the CodeWatch main window. A document window cannot be moved outside the boundaries of the frame of the main window but can be positioned anywhere within the frame of the main window not occupied by a dockable window.</p> <p>When a document window is maximized to occupy the entire document area, the title bar is hidden.</p> <p>The tab at the bottom of a document window allows a window to be quickly chosen. A traditional MDI window style can be selected by changing the <b>Window Style</b> on the <b>Edit→View Preferences→Miscellaneous</b> tab.</p> <p>For more information, see <a href="#">Document Windows</a> (on page 59).</p>
<b>Floating Windows</b>	<p>Floating windows are similar in appearance to dockable windows, but cannot be docked and always float free of the frame of the CodeWatch main window. Floating windows are used by the Find command on the Edit menu and the Data Item command on the View menu.</p> <p>For more information, see <a href="#">Floating Windows</a> (on page 62).</p>

The following sections describe the content of the CodeWatch main window.

---

## Dockable Windows

Three of the windows used in CodeWatch are dockable. These include the following:

- [Workspace window](#) (see the following topic)
- [Data window](#) (see page 51)
- [Log window](#) (see page 58)

As discussed in [Table 1](#) on page 46, CodeWatch allows you to convert any dockable window into a document window. To do so, click in the window you wish to convert, and uncheck **Docked** on the **Window** menu. This option is also available on the pop-up context menu that appears if you right-click in the window or on the title bar of any of the dockable windows.

## Workspace Window

The Workspace window is initially docked to the left side of the frame of the CodeWatch main window, and allows you to examine and manipulate the contents of the workspace. The Workspace window is always visible when a workspace is active. By clicking on the Workspace window's title bar, you can reposition it anywhere within the CodeWatch main window. You can close the Workspace window at any time by clicking the  **Close** button at the end of the window's title bar.

The Workspace window includes two tabs:

- [Programs tab](#) (see page 48)
- [Files tab](#) (see page 49)

To see the workspace organized by programs, click the **Programs** tab. To see the workspace organized by files, click the **Files** tab. If an item is preceded by a [+] symbol, the item is expandable. Click [+] in front of an item to display the contents of that item. If an item is preceded by a [-] symbol, click [-] to minimize the size of the list.

The following icons are used in the Workspace window:

Workspace Window	
This icon	Represents
	A grouping of items. Note that this icon does not represent a directory on disk.
	An RM/COBOL source file. When the cog portion of this icon is red, the source file has been modified more recently than the program file.
	An RM/COBOL COPY file. When the cog portion of this icon is present, the COPY file has been used to compile any of the programs in the workspace. When the cog portion of this icon is red, the COPY file has been modified more recently than any of the program files.
	In order shown, an RM/COBOL program: without source, with source, and with source that has been modified since the last compilation.
	An RM/COBOL program file or library file. When the gear portion of this icon is red, one of the programs within the program file or library file is out-of-date either because a source file or a COPY file has been edited or is missing.

Closing the Workspace window also closes the workspace. CodeWatch will prompt you to save the workspace if the workspace has not been saved, or if changes have been made since the last save. You can then save or discard the changes before closing the workspace, or cancel the operation.

## Programs Tab

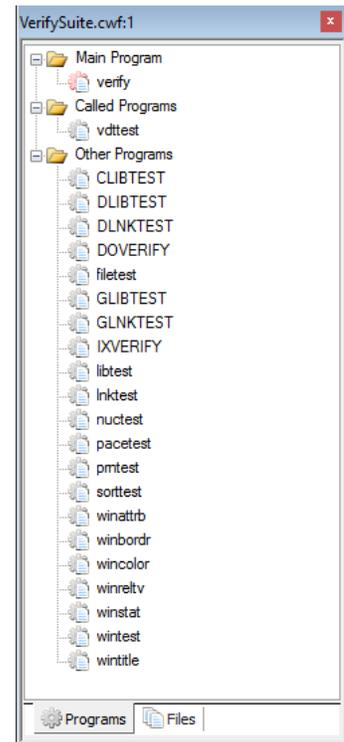
The **Programs** tab in the Workspace Window, illustrated to the right, contains the names of all of the programs in the workspace, organized into three sections:

- **Main Program.** Contains the name of the main RM/COBOL program. If the name has not been resolved, the name is enclosed in <angle brackets> to indicate that it is a placeholder. As depicted here, the icon is red if the file has been edited.
- **Called Programs.** When debugging, contains a list of programs that have been called during the current run. The currently active program is always at the top of the list.
- **Other Programs.** Contains a list of the programs that have not yet been called in the current run.

A new workspace contains no programs, other than a placeholder for the main program. When the workspace is run for the first time, the runtime system resolves the program name to an actual program and replaces the placeholder. As additional programs are called during a debugging session, the names of these programs are added to the Called Programs folder, which is a convenient list of programs that have been invoked during the session. At the beginning of the next debugging session, the called programs are moved into the Other Programs folder, and migrate into the Called Programs folder as they are again invoked.

The list of programs is saved when you save the workspace to a file. This makes it possible to visually browse source files and set breakpoints without first having to run the program to recreate the Called Programs folder. To see the source code for a given program, double-click on the name of the program. The source code opens in a [Source Code/Edit window](#) (see page 97).

**Note** In addition to the drop-down menus on the menu bar, CodeWatch provides context-sensitive menus that pop up at the click of the right mouse button to access frequently-performed operations quickly. The contents of the context menu changes depending on the item that is currently selected. To open a context menu in the Workspace window, select an object and right-click. From the pop-up context menu, choose the command you need. To learn more about the context menu for the Programs tab, see [Context Menu in a Workspace Window](#) (on page 49).



## Files Tab

The Files tab in the Workspace Window, illustrated to the right, contains the names of all of the files in the workspace, organized into four sections:

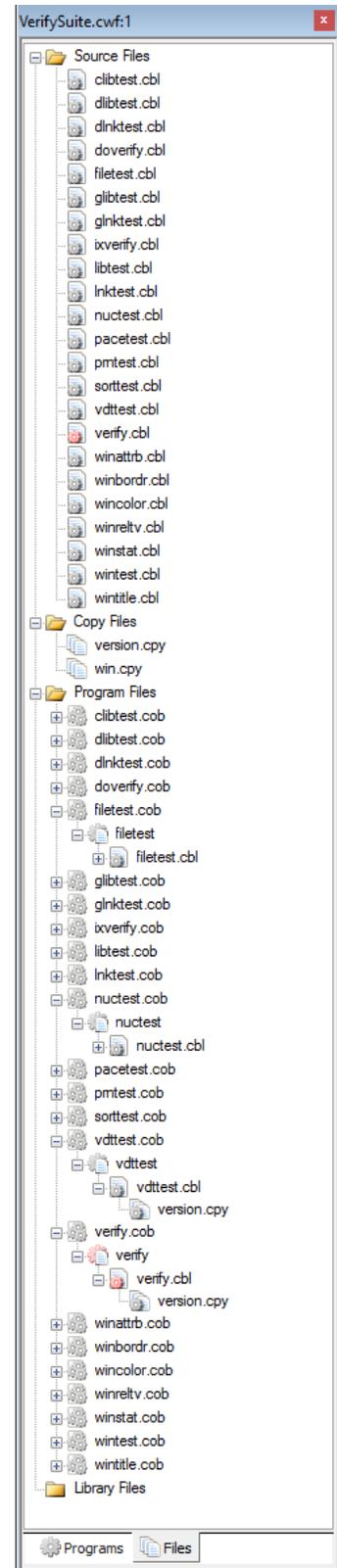
- **Source Files.** Contains the list of source files in the workspace. Source files that are inserted into this list as a result of a reference from a program are displayed using a green icon. Source files that are inserted manually using the Open or Insert Items commands from the File menu appear gray.
- **Copy Files.** Lists all copy files in the workspace. Copy files that are inserted into this list as a result of a reference from a program are displayed using a green icon. Copy files that are inserted manually using the Open or Insert Items commands from the File menu appear gray.
- **Program Files.** Contains the RM/COBOL program files, by name, that are run.
- **Library Files.** Contains the RM/COBOL library files that were specified using the /L Runtime Command Option or that were added to the workspace library list.

The Program Files and Library Files folders can be expanded to reveal the list of program or library files in each folder. Each program file or library file can be further expanded to reveal the list of programs within the file, as well as the source and copy files that were used to compile the program file or library file. Programs that contain other programs can be expanded to reveal the nesting relationship between programs. Finally, double-clicking or pressing F2 on any of these elements will open the source file for the program in a [Source Code/Edit window](#) (see page 97) for examination.

**Note** In addition to the drop-down menus on the menu bar, CodeWatch provides context-sensitive menus that pop up at the click of the right mouse button to access frequently-performed operations quickly. The contents of the context menu changes based on the currently selected item. To open a context menu in the Workspace window, select an object and right-click. From the pop-up context menu, choose the command you need. To learn more about the context menu for the Files tab, see [Context Menu in a Workspace Window](#) (on page 49).

## Context Menu in a Workspace Window

When you right-click over a selected object in either the Programs or Files tab of a Workspace window, a pop-up context menu appears. The contents of the context menu will vary depending on the currently selected object and the currently



selected tab. The following table displays the possible context-menu commands and their purpose. See also [Menus and Toolbars](#) (on page 68).

Context Menu in a Workspace Window		
Command	Shortcut Key	Action
<b>Insert Source File</b> <b>Insert Copy File</b> <b>Insert Program File</b> <b>Insert Library File</b>	Ins	<p>Displays the Open File dialog box and allows you to choose one or more files to be added to the workspace.</p> <p>The type of item added by this menu command is determined by the currently highlighted item in the Workspace window. For example, to insert a source file, right-click on an existing source file or the Source File folder in the Files tab.</p> <p>You can insert any number of files in one operation by holding down the Shift key when selecting files in the Open File dialog box. All files must be the same type.</p> <p>Note that the menu command displayed will reflect the action that will be performed.</p>
<b>Remove “filename”</b>	Del	<p>Removes the specified item from the workspace. The type of item that is removed is determined by the selection. For example:</p> <ul style="list-style-type: none"> <li>• Folders cannot be removed from the workspace.</li> <li>• Source and copy files can be removed from the workspace only if they are not in use by a COBOL program or library. In that case, you must instead remove the program or library from the workspace.</li> </ul> <p>Note that the text of the menu command will change to reflect the action that will be performed.</p>
<b>Compile File</b>	(Ctrl+F7)	<p>Compiles the selected item, as follows:</p> <ul style="list-style-type: none"> <li>• If the focus is in the Workspace window, and a COBOL source file is selected, that file is compiled. In that case, the menu command will contain the name of the file to compile.</li> <li>• If the source file is open in a Source Code/Edit window and has been edited and not saved, you are prompted to save the file or cancel the compilation.</li> </ul>

Context Menu in a Workspace Window		
Command	Shortcut Key	Action
<b>Properties</b>	Alt+Enter	<p>Displays the properties of the selected item in the Workspace Properties dialog box, as follows:</p> <ul style="list-style-type: none"> <li>• The dialog box that is displayed is dependent on the item that is currently selected in the Workspace window.</li> <li>• The item properties that can be examined include the following: <ul style="list-style-type: none"> <li>- a list of included copy files;</li> <li>- a list of programs produced when the source file was compiled or a list of source files that depend on the copy file;</li> <li>- for source files, the build settings that were in effect for the last compilation.</li> </ul> </li> <li>• The Source tab on the Environment tab of the Workspace Properties dialog box allows you to set the compilation options for the specified item, and allows you to set the default compilation options to be used for all files that do not have individual settings.</li> </ul> <p>For a complete discussion, see the <a href="#">Inspect/Modify dialog box</a> (on page 66).</p>
<b>Edit File “name” or Edit Source</b>	F2	Opens the specified item in the Source Code/Edit Window. This operation may be performed on program files, programs, source files, and copy files.
<b>Docked</b>		Checked (✓), if the window is docked to the frame of the CodeWatch main window. Deselecting this menu item converts the window from a dockable window into a document window. Checking this item docks the window to the frame of the main window.
<b>Floating</b>		Checked (✓), if the window is floating free of the frame of the CodeWatch main window. Deselecting this menu item docks the window; checking this window causes it to float free of the frame in the main window.

## Data Window

The Data window, illustrated below, displays information about the program that you are debugging. This window contains a set of five tabs:

- [Breakpoints tab](#) (see page 52)
- [Watchpoints tab](#) (see page 53)
- [Data Files tab](#) (see page 54)
- [Call Stack tab](#) (see page 55)
- [Build Results tab](#) (see page 56)



To select a tab in the Data window, click it. Alternatively, from the **View** menu, select the command or press the shortcut key that corresponds to the tab you want to display.

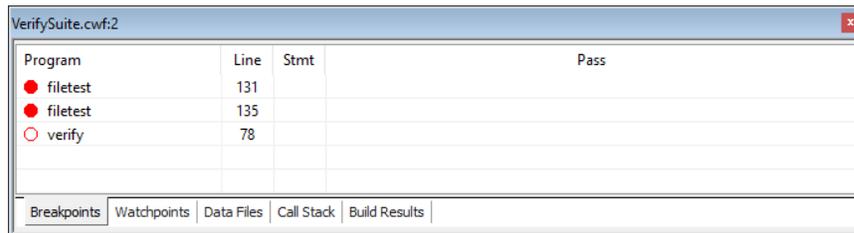
The Data window is opened automatically whenever you click the **Start** command from the **Run** menu or click the **Run** toolbar button. You can close the Data window at any time by clicking the **Close** button at the end of the window's title bar.

To reopen the Data window, choose either **Breakpoints Tab**, **Watchpoints Tab**, **Data Files Tab**, **Call Stack Tab**, or **Build Results Tab** from the **View** menu or press the corresponding shortcut key.

## Breakpoints Tab

To access the Breakpoints tab, illustrated below, do one of the following:

- Click the **Breakpoints** tab in the Data window.
- Choose **Breakpoints Tab** from the **View** menu.
- Press **Alt+1**.



Breakpoints Tab	
This option	Describes
<b>Program</b>	The name of the program containing the breakpoint. The Debugger does not differentiate between separately compiled programs and contained (that is, nested) programs that have the same name. Execution will stop when any instance of the named program is entered and the specified line is encountered. The program name is preceded by a red circle if the breakpoint is enabled, or by a hollow circle if the breakpoint is temporarily disabled.
<b>Line</b>	The number of the target line.
<b>Stmt</b>	The statement number specified within the line. Leave this blank to indicate that the break will occur when the first (or only) statement on the line is executed. Or, specify an offset: +1 indicates the second statement on the line, and +2 indicates the third.
<b>Pass</b>	The number of times a breakpoint should be skipped before being triggered. The default is to trigger the breakpoint on each pass. For example, a value of 5 indicates that the breakpoint will be skipped four times and then triggered. Once a breakpoint is triggered, it will be triggered on each subsequent pass unless the pass count is set again.

You can perform the following operations in the Breakpoints tab:

- Enable or disable a breakpoint by pressing the Spacebar or by single-clicking the red circle preceding the program name. When enabled, the program will halt when the

conditions attached to the breakpoint are satisfied. When disabled, the breakpoint has no effect on the program.

- Add a new breakpoint by pressing the **Ins** key.
- Remove a breakpoint by pressing the **Del** key.
- Edit a breakpoint by pressing the **F2** key.
- Display a pop-up context menu by right-clicking with the mouse (alternatively, press **Shift+F10**). This pop-up menu allows you to change the properties of the Breakpoints tab using the Code Breakpoints tab of the [Breakpoints & Watchpoints dialog box](#) (see page 87), as well as perform the other operations listed above.

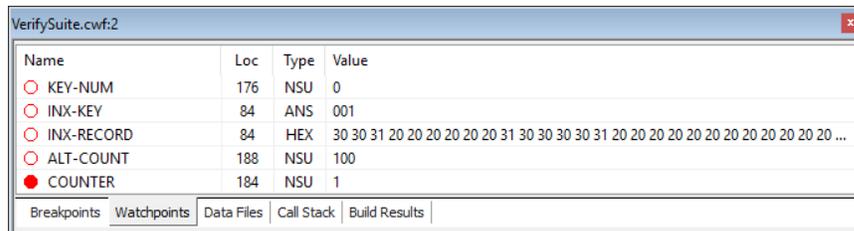
You can also set, clear, enable, or disable a breakpoint in the [Source Code/Edit window](#) (see page 97) by one of the following methods:

- Double-click on the line number to set or clear a breakpoint.
- Right-click on a line to set, clear, enable, or disable a breakpoint by using the options on the pop-up context menu.

## Watchpoints Tab

To access the Watchpoints tab, illustrated below, do one of the following:

- Click the **Watchpoints** tab in the Data window.
- Choose **Watchpoints Tab** from the **View** menu.
- Press **Alt+2**.



Watchpoints Tab	
This option	Describes
<b>Name</b>	The name of the data item being watched. The name is preceded by a red circle to halt the program whenever the content of the data item changes. A hollow circle indicates that the watchpoint will not halt the program.
<b>Loc</b>	The location of the data item within the program.
<b>Type</b>	The type of the data item. This field determines how the value is displayed. It defaults to the actual type of the data item, but may be overridden using the <a href="#">Inspect/Modify dialog box</a> (see page 66). For example, INX-RECORD above was changed to display in HEX.
<b>Value</b>	The value of the data item.

You can perform the following operations in the Watchpoints tab:

- Enable or disable a watchpoint by pressing the Spacebar or by single-clicking the red circle preceding the data item name. When enabled, execution halts whenever the value in the data item changes. When disabled, the value is updated whenever the program halts, but a value change will not cause the program to halt.
- Add a new watchpoint by pressing the **Ins** key.
- Remove a watchpoint by pressing the **Del** key.
- Edit a watchpoint or modify the value of the data item by pressing the **F2** key.
- Display a pop-up context menu by right-clicking with the mouse or by pressing **Shift+F10**. This pop-up menu allows you to change the properties of the watchpoints listed on the Watchpoints tab using the Data Watchpoints tab of the [Breakpoints & Watchpoints dialog box](#) (see page 87), as well as perform the other operations, as described above.

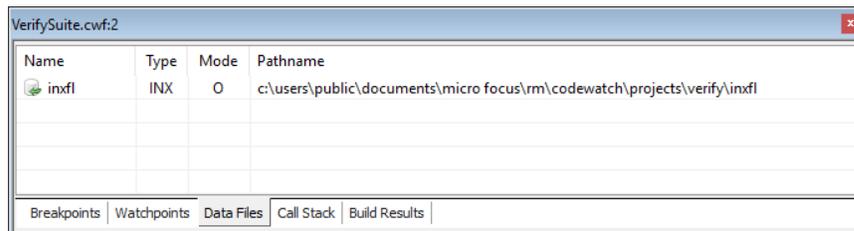
You can also add a watchpoint in the [Source Code/Edit window](#) (see page 97) by using one of the following methods:

- Select the name of a data item with the mouse, double-click to inspect the data item, and then click the **Watch** button in the [Inspect/Modify dialog box](#) (see page 66). (Note that you can also modify the data item by adding qualification or subscribing in this way. See the *RM/COBOL User's Guide* for more information.)
- Select the name of a data item with the mouse, then right-click and choose **Watch Data Item** from the pop-up context menu.

### Data Files Tab

The Data Files tab, illustrated below, contains a running trace of all files opened during the execution of this program and is for information purposes only. To access the Data Files tab, do one of the following:

- Click the **Data Files** tab in the Data window.
- Choose **Data Files Tab** from the **View** menu.
- Press **Alt+3**.



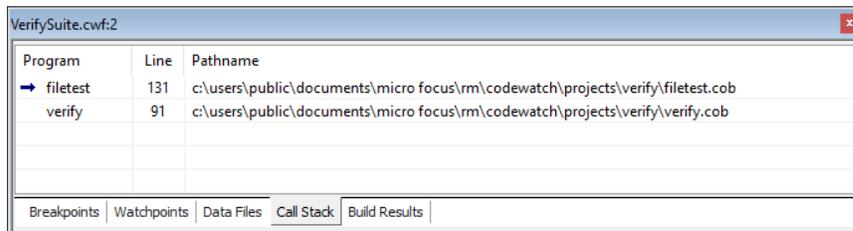
Data Files Tab	
This option	Describes
Name	The name of the file.

Data Files Tab	
This option	Describes
<b>Type</b>	The type of file, where: <ul style="list-style-type: none"> <li>• PGM Indicates program file.</li> <li>• SEQ Indicates sequential file.</li> <li>• REL Indicates relative file.</li> <li>• INX Indicates indexed file.</li> </ul>
<b>Mode</b>	The mode used to open the file, where: <ul style="list-style-type: none"> <li>• I Indicates open for input.</li> <li>• IO Indicates open for input and output.</li> <li>• O Indicates open for output.</li> <li>• E Indicates open for extend (output at end of file).</li> </ul>
<b>Pathname</b>	The full pathname of the file.

### Call Stack Tab

The Call Stack tab, illustrated below, contains a list of currently active programs and the number of the currently active line within each of those programs. To access the Call Stack tab, do one of the following:

- Click the **Call Stack** tab in the Data window.
- Choose **Call Stack Tab** from the **View** menu.
- Press **Alt+4**.



Call Stack Tab	
This option	Describes
<b>Program</b>	The name of the program. The currently active program is always indicated by the blue arrow and is displayed at the top of the list. The next line is the caller of the currently active program, and the next line is its caller.  Note that you can return to the current program's caller by choosing Step Program from the <a href="#">Run menu</a> (see page 77).
<b>Line</b>	The number of the line currently executing in each program.
<b>Pathname</b>	The filename of the COBOL program file or library that contains each program.

You can perform the following operations in the Call Stack tab:

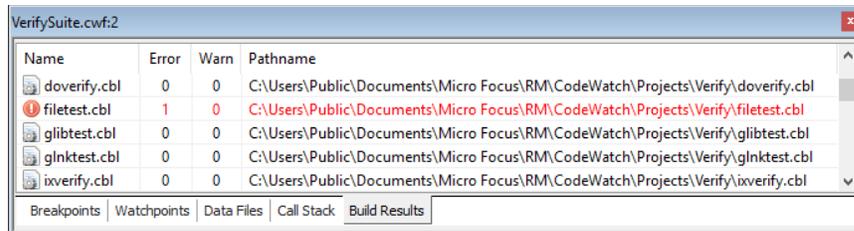
- Open a [Source Code/Edit window](#) (see page 97) and move to the indicated line by pressing **Enter** or **F2**.
- Display a pop-up context menu by right-clicking with the mouse, or by pressing **Shift+F10**. This pop-up menu allows you to change the properties of the Call Stack tab.

**Note** You can see a list of all programs that have been executed in this session by examining the [Programs tab](#) (see page 48) in the Workspace window.

## Build Results Tab

The Build Results tab, illustrated below, contains a list of the source files compiled by the last build, along with the results of each compilation. Note that if any errors or warnings were detected, the line is displayed in red. To access the Build Results tab, do one of the following:

- Click the **Build Results** tab in the Data window.
- Choose **Build Results Tab** from the **View** menu.
- Press **Alt+5**.



Build Results Tab	
This option	Describes
<b>Name</b>	The name of the file. A red exclamation mark is displayed if the file compiled with errors (indicating recompilation is required), and the entire row is displayed in red if there were any errors or warnings.
<b>Error</b>	The number of errors detected during the compilation.
<b>Warn</b>	The number of warnings detected during the compilation.
<b>Pathname</b>	The full pathname of the file.

You can perform the following operations in the Build Results tab:

- Open a [Source Code/Edit window](#) (see page 97) and move to the indicated line by pressing **Enter** or **F2**.
- Display a pop-up context menu by right-clicking with the mouse, or by pressing **Shift+F10**. This pop-up menu allows you to change the properties of the Build Results tab.

## Context Menu in a Data Window

When you right-click in a Data window, a pop-up context menu appears. The contents of the context menu will vary depending on the currently selected item and the currently selected tab. The following table displays the possible context-menu commands and their respective purposes. See also [Menus and Toolbars](#) (on page 68).

Context Menu in a Data Window		
Command	Shortcut Key	Action
<b>New</b>	Ins	Displays either the Code Breakpoints tab or the Data Watchpoints tab of the <a href="#">Breakpoints &amp; Watchpoints dialog box</a> (see page 87) and allows you to manually define an item. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Delete</b>	Del	Deletes the specified breakpoint or watchpoint. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Properties</b>	Alt+Enter	Displays the properties for the selected item and allows you to edit the properties in either the Code Breakpoints tab or the Data Watchpoints tab of the <a href="#">Breakpoints &amp; Watchpoints dialog box</a> (see page 87). This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Edit Source</b>	F2	For breakpoints, the call stack, or build results, edits the source file referenced by the selected item and moves the cursor to the referenced line. This option is not enabled in the Watchpoints or Data Files tabs.
<b>Enabled</b>	Spacebar	For breakpoints and watchpoints, enables or disables the selected item. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Clear All</b>		For breakpoints and watchpoints, removes the selected item. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Disable All</b>		Disables all breakpoints or watchpoints. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Enable All</b>		Enables all breakpoints or watchpoints. This option is not enabled in the Data Files, Call Stack, or Build Results tabs.
<b>Docked</b>		Checked (✓), if the Data window is docked to the frame of the CodeWatch main window. Deselecting this menu item converts the window from a dockable window into a document window. Checking this item docks the window to the frame of the main window.
<b>Floating</b>		Checked (✓), if the Data window is floating free of the frame of the CodeWatch main window. Deselecting this menu item docks the window; checking this window causes the Data window to float free in the frame in the main window.



## Log Window

The Log window, illustrated below, is opened automatically whenever you click the **Start** command from the **Run** menu or click the **Run** toolbar button. You can also access the Log window by choosing **Log Window** from the **View** menu or pressing **Alt+6**.

```
VerifySuite.cwf:3
Loading RM/COBOL application...
Working directory is "C:\Users\Public\Documents\Micro Focus\RM\CodeWatch\Projects\Verify"
Running "VERIFY -D"
ST 44 verify C? B 131,filetest
44 verify C? B 135,filetest
44 verify C? B 70,verify
44 verify C? B 92,verify
44 verify C? @S999
BP 92 verify C? @D FUNCTION-NUMBER
S2 NSU 2
92 verify C?
<
```

The Log window serves two purposes:

- Displays compiler messages during and after a compilation. The window is automatically cleared before each command on the [Build menu](#) (see page 76) is executed.
- Allows you to directly issue commands to the RM/COBOL command line Debugger during program execution, as shown in the example above. Any of the commands, described in Chapter 9: *Debugging* of the *RM/COBOL User's Guide*, can be entered directly in the Log window. The Debugger's output is also displayed in this window.

To enter a Debug command, click anywhere within the Log window, type the command, and press **Enter**. To view previous commands or output, scroll through the window by using the scroll bars. Note that the Log window displays only the most recent 100 lines of output.

You can close the Log window at any time by clicking the **Close** button at the end of the window's title bar. To reopen the Log window, choose **Log Window** from the **View** menu or press **Alt+6**. The Log window is also opened automatically whenever you click the **Start** command from the **Run** menu.

### Context Menu in a Log Window

When you right-click in a Log window, a pop-up context menu appears. The contents of the context menu will vary depending on the currently selected item. The following table displays the possible context-menu commands and their purpose. See also [Menus and Toolbars](#) (on page 68).

Context Menu in a Log Window	
Command	Action
<b>Clear</b>	Clears the contents of the Log window.
<b>Docked</b>	Checked (✓), if the Log window is docked to the frame of the CodeWatch main window. Deselecting this menu item converts the window from a dockable window into a document window. Checking this item docks the window to the frame of the main window.

Context Menu in a Log Window	
Command	Action
<b>Floating</b>	Checked (✓), if the Log window is floating free of the frame in the CodeWatch main window. Deselecting this menu item docks the window; checking this window causes it to float free of the frame in the main window.

## Document Windows

The area within the frame of the CodeWatch main window that is not occupied by the [dockable windows](#) (see page 46) is reserved for “document” windows. Document windows have a large title bar, are listed in the Window menu, and always float only in the document area of the main window. (If the document window’s title bar is not visible, click the  **Restore** button.) Unlike dockable windows, you can have any number of document windows that are displaying different objects. You can even display the same object in two different document windows. The [Source Code/Edit window](#) (see page 97) is an example of a document window.

**Note** CodeWatch allows you to convert any dockable window into a document window. To do so, click in the window you wish to convert, and uncheck **Docked** on the **Window** menu. This option is also available on the pop-up context menu that appears if you right-click in the window or on the title bar of the dockable window.

### Using Document Windows

The sections in this topic:

- Describe the state of document windows
- Detail the Document menu commands and toolbar buttons
- Explain how to cycle through, split, and close document windows

#### **State of Document Windows**

Document windows can take on three different states: minimized, maximized, or normal.

- **Normal.** In the normal state, all document windows are overlapped in the document area. You can use commands on the Window menu to arrange all of the document windows, either by tiling or cascading the windows.
- **Maximized.** In the maximized state, only the topmost document window is visible and there is no separate document title bar. You can move between document windows by pressing **Ctrl+Tab** or by using the Window menu.
- **Minimized.** In the minimized state, the document window is reduced so only its title bar is visible at the bottom of the document area.

The state of a document window is set by one of the following methods:



- Using the Document menu that appears when you click the document icon in the upper-left corner of the title bar. If the document window's title bar is not visible, restore the document by clicking the **Restore** button.
- By clicking the **Minimize**, **Maximize**, or **Close** buttons on the upper-right corner of the title bar.

**Note** If the document window is maximized, the Document menu icon is moved to the left side of the menu bar and the buttons are moved to the right side of the menu bar.

To activate the Document menu from the keyboard:

1. Press the **F10** key, or press and release the **Alt** key. This activates the **File** menu.
2. Press the Left Arrow key, which activates the Document menu for the current window.
3. Press the Down Arrow key, which drops down the menu.

## Document Menu

The Document menu contains the following commands. The corresponding shortcut key and button, if applicable, are also shown.

Document Menu			
Command	Shortcut Key	Button	Action
<b>Restore</b>			<p>Restores the views so all document windows are potentially visible in the document area. The Restore button always replaces the Maximize button or the Minimize button when that command is carried out.</p> <p>Use the following commands to arrange the restored windows:</p> <ul style="list-style-type: none"> <li>• <b>Window   Cascade</b> to cascade all document windows from left to right, top to bottom across the document area.</li> <li>• <b>Window   Tile</b> to tile all document windows in the document area so a part of each window is visible.</li> </ul>
<b>Move</b>			<p>Moves the window. You can use the mouse or the arrow keys to position the window in the document area. This function is not available if the window is maximized.</p> <p>You can also move a window with the mouse by clicking anywhere in the title bar and dragging the window to a new location.</p>
<b>Size</b>			<p>Resizes the window. The Up and Down Arrow keys, respectively, increase and decrease the size of the window vertically, and the Left and Right Arrow keys perform the same operation horizontally. This function is not available if the window is maximized or minimized.</p> <p>You can also size a window with the mouse by clicking the frame on any side, or clicking a corner and dragging with the mouse.</p>

Document Menu			
Command	Shortcut Key	Button	Action
<b>Maximize</b>			Maximizes all document windows so the top-most window occupies the entire document area. No title bar is visible, the document menu is moved to the left side of the menu bar, and the document window buttons are moved to the right end of the menu bar.
<b>Minimize</b>			Minimizes the currently selected document window. A minimized window is represented by a title bar at the bottom of the document area. Note that the title bar is not visible if another maximized document window is available.
<b>Close</b>	Ctrl+F4		Closes the currently active document window, regardless of the state of the window. If the document has been changed but not saved, you will be prompted to save or discard the changes.
<b>Next</b>	Ctrl+F6 or Ctrl+Tab		Activates the next document window.
<b>Previous</b>	Ctrl+Shift+F6 or Ctrl+Shift+Tab		Activates the previous document window. (Note that this command is not on the menu).

When CodeWatch is closed, it remembers the state of the last document window (maximized or normal), and that setting is used the next time CodeWatch is started.

### Cycling Through Open Windows

Document windows are kept in a stack. If document windows are maximized, you can see only the top-most document in the stack. To cycle through all open document windows, do one of the following:

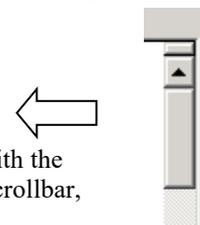
- Press **Ctrl+F6** or **Ctrl+Tab** to move forward to the next window in the document window stack.
- Press **Ctrl+Shift+F6** or **Ctrl+Shift+Tab** to move back to the previous window in the document window stack.

### Splitting Document Windows

CodeWatch allows you to split a document window into either two or four smaller windows (called panes) that can be independently positioned over the text. This is useful for viewing the Working-Storage Section of your source at the same time as viewing the Procedure Division.

There are two ways to split a document window:

- From the **Window** menu, choose **Split**.
- Alternatively, drag the **split box** (illustrated to the right) with the mouse. The split boxes are located on top of the vertical scrollbar, and to the left of the horizontal scrollbar.



In either case, the document window is immediately transected by a horizontal and a vertical divider (called a **split bar**), and the intersection of the dividers may be moved with the mouse or the arrow keys on the keyboard. Press the Enter key or release the mouse button when you are finished positioning the split bar.

When you split document windows, there are several considerations to keep in mind:

- To divide the window into two panes instead of four, drag the unwanted split bar to the edge of the window.
- The panes may be positioned independently, but only the pane containing the cursor is active.
- To move the cursor into the next pane, choose **Pane | Next** from the **Window** menu or press **F6**. To move the cursor into the previous pane, choose **Pane | Previous** from the **Window** menu or press **Shift+F6**.

**Note** Neither of the **Pane** commands will activate a different document window. To activate a different document, press **Ctrl+F6** or **Ctrl+Shift+F6**.

- To adjust the split bar, choose **Split** from the **Window** menu. To move the divider with the mouse, simply click and drag the split bar to the new position.
- To close a pane, drag the split bar to the edge of the window with the keyboard or the mouse, or double-click the divider with the mouse.

## Closing Document Windows

Closing a document window returns you to the previous document window in the stack. There are several ways to close a document window. Do one of the following:

- Press **Ctrl+F4** or click the **Close** button on the document window's title bar. If the window is maximized, the Close button is at the right end of the menu bar.
- Choose **Close** from the Document menu.

These functions close only the currently active document window and close the document if there is only one open document window. If you used the **New Window** command from the **Window** menu to create additional windows on the document, the document is not closed until all windows are closed.

To close all windows associated with the current document:

- Choose **File | Close**.

If the document has been changed, you will be given the option to save the document.

---

## Floating Windows

Floating windows “float” over the CodeWatch main window and may be revealed or hidden at any time. Three of the dialog boxes used in CodeWatch are floating type windows:

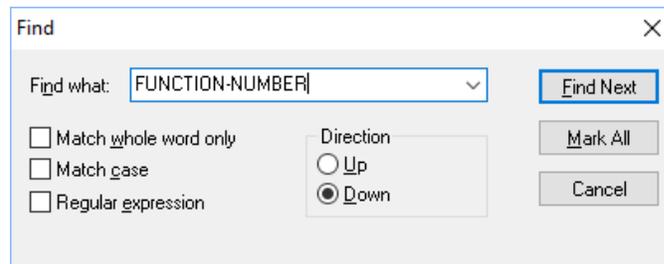
- [Find dialog box](#) (see the following topic)
- [Replace dialog box](#) (see page 65)
- [Inspect/Modify dialog box](#) (see page 66)

## Find Dialog Box

The Find dialog box, illustrated below, allows you to search for text in the source file in various ways. This dialog box is enabled only when a [Source Code/Edit window](#) (see page 97) is open.

To display the Find dialog box, do one of the following:

- From the **Edit** menu, choose **Find**.
- Click the **Find** toolbar button.
- Press Ctrl+F or Alt+F3.



Find Dialog Box	
Use this option	To
<b>Find what</b>	For the current file, type the information you want to search for (this is referred to as the search criteria). Or, click a previous search entry from the list.
<b>Match whole word only</b>	Match the search string only if the found occurrence is delimited by: <ul style="list-style-type: none"> <li>• spaces, tabs, or the beginning and/or end of the line, or</li> <li>• any of the following characters:  \ " ' ` ~ - = ! @ # \$ % ^ &amp; *  ( ) + [ ] { }   ; : . , &lt; &gt; / ?</li> </ul> For example, searching for “25” when this option is not checked will match “1255” and “A25”. Checking this option ensures that only “25” is matched.
<b>Match case</b>	When selected, find only the text that exactly matches the search text criteria. When not selected, uppercase and lowercase letters are considered to be identical. Therefore, searching for ABC will match “abc” and “AbC” in addition to “ABC”.
<b>Regular expression</b>	When selected, specify that the search text is a UNIX-style regular expression. This provides the ability to perform powerful search operations. For more information, see <a href="#">Regular Expressions</a> (on page 64).
<b>Direction</b>	Control the direction of the search: select Down to search forward in the file toward the end, or select Up to search backward in the file toward the beginning.

Find Dialog Box	
Use this option	To
<p><b>Find Next</b></p>  	<p>In the current file, find the next occurrence of the text string entered or selected in the Find what list box in the specified direction. If found, the text is highlighted.</p> <p>Pressing F3, choosing Find Next from the Edit menu, or clicking the Find Next toolbar button performs this same operation using the current settings in the Find dialog box, even if the Find dialog box is not open (or visible).</p> <p>Find the previous occurrence of the text in the current file. If found, the text is highlighted. Note that pressing Shift+F3, choosing Find Previous from the Edit menu, or clicking the Find Previous toolbar button performs this operation using the current settings in the Find dialog box, even if the window is not visible.</p>
<b>Mark All</b>	Mark all the found occurrences.
<b>Cancel</b>	Close the Find dialog box without matching and cancel all entries made in the dialog box.

## Regular Expressions

CodeWatch allows you to search for text patterns, called regular expressions, rather than fixed strings. You indicate that the search string is a regular expression by checking the Regular expression check box in either the [Find dialog box](#) (see page 63) or the [Replace dialog box](#) (see page 65).

Regular expressions are made by combining normal characters with a number of special characters. For example:

```
^b*88b*[FfG-I]
```

where **b** is a space that matches “88,” followed by one or more spaces, followed by any of the letters in the string “FfGHI”. The 88 must be the first non-blank characters on the line.

[Table 2](#) lists these special characters and their uses as defined by CodeWatch.

**Table 2: Special Characters Allowed in CodeWatch Regular Expressions**

Special Characters Allowed in CodeWatch Regular Expressions	
Special Character	Description
^ (caret)	Indicates that the regular expression immediately following this character must be found at the beginning of the line.
\$	Indicates that the regular expression immediately preceding this special character must be found at the end of the line.
. (period)	Matches a single character.
*	Matches any number of the character or pattern that immediately precedes it.

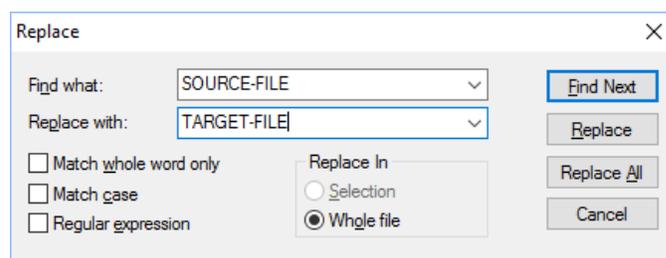
Special Characters Allowed in CodeWatch Regular Expressions	
Special Character	Description
[...]	Matches any one of the characters or a range of consecutive characters enclosed between brackets. For example, [xy] matches either x or y. A range of consecutive characters is specified by separating the first and last characters in the range with a hyphen. For example, [0-9] will match any digit from 0 to 9.  <b>Note</b> A caret (^) changes its special meaning inside brackets. A caret as the first character in the bracket reverses its meaning; it tries to match any character not in the list. To match a circumflex, place it later in the range.
\{n,m\}	Matches a range of occurrences of the single character (including a character specified by a regular expression) that immediately precedes it. <i>N</i> and <i>m</i> are integers between 0 and 256 that specify the number of occurrences to match. \{n\} matches exactly <i>n</i> occurrences, \{n,\} matches at least <i>n</i> occurrences, and \{n,m\} matches any number of occurrences between <i>n</i> and <i>m</i> . For example, L\{2,3\} will match either LL (as in LLAMA) or LLL (as in LLL Ranch), but not the single letter L.
\ (backslash)	Indicates that the following special character is treated as an ordinary character; for example, “\.” stands for a period and “\*” stands for an asterisk.

## Replace Dialog Box

The Replace dialog box, illustrated below, allows you to search for and replace text in the source file in various ways. This window is enabled only when a [Source Code/Edit window](#) (see page 97) is open.

To reveal or hide the Replace dialog box, do one of the following:

- From the **Edit** menu, choose **Replace**.
- Press **Ctrl+H**.



Replace Dialog Box	
Use this option	To
<b>Find what</b>	For the current file, enter or select the text to search for. (You can click the down-arrow to see a list of previous searches and repeat a search by clicking in the list.)
<b>Replace with</b>	Enter or select the text that is to replace each matched instance of the search criteria specified in the Find what option.

Replace Dialog Box	
Use this option	To
<b>Match whole word only</b>	<p>Match the search criteria specified in the Find what option only if the found occurrence is delimited by:</p> <ul style="list-style-type: none"> <li>spaces, tabs, or the beginning and/or end of the line, or</li> <li>any of the following characters:  <code>\ " \ ` ~ - = ! @ # \$ % ^ &amp; *            ( ) + [ ] { }   ; : . , &lt; &gt; / _</code> </li> </ul> <p>For example, searching for "25" when this option is not selected will match "1255" and "A25". Selecting this option ensures that only "25" is matched.</p>
<b>Match case</b>	<p>When selected, find only the text that exactly matches the search criteria. When not selected, uppercase and lowercase letters are considered to be identical. Therefore, searching for ABC will match "abc" and "AbC" in addition to "ABC".</p>
<b>Regular expression</b>	<p>When selected, specify that the search criterion is a UNIX-style regular expression. This provides the ability to perform powerful search and replace operations. For more information, see <a href="#">Regular Expressions</a> (on page 64).</p>
<b>Replace In</b>	<p>Limit the scope of the Replace All operation (discussed below). You can choose to replace all matching instances of the search criteria specified in the Find what option in either the selection or the entire file.</p>
<b>Find Next</b>	<p>Find and highlight the next occurrence of the search criteria specified in the Find what option in either the selection or the entire file.</p> <p>Pressing F3, choosing Find Next from the Edit menu, or clicking the Find Next toolbar button performs this same operation using the current settings in the Replace dialog box, even if the Replace dialog box is not open (or visible).</p> <p>Find the previous occurrence of the text in the current file. If found, the text is highlighted. Note that pressing Shift+F3, choosing Find Previous from the Edit menu, or clicking the Find Previous toolbar button performs this operation using the current settings in the Find dialog box, even if the dialog box itself is not visible.</p>
<b>Replace</b>	<p>Replace the currently highlighted instance of the search criteria in the file with the replacement string. If you want to replace all occurrences of the search criteria automatically, choose the Replace All option.</p>
<b>Replace All</b>	<p>Replace all instances of the search criteria in the file with the replacement string. Compare with the Replace option.</p>
<b>Cancel</b>	<p>Terminate the Replace dialog box without matching and cancel all entries made in the dialog box.</p>

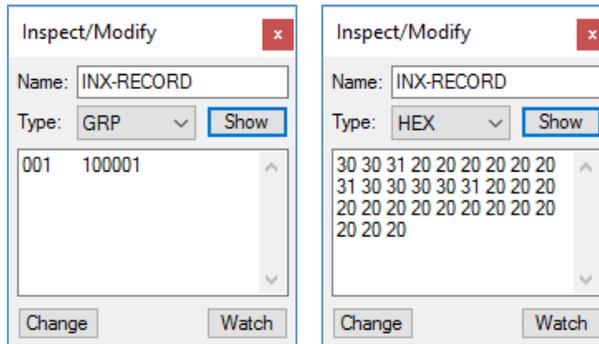
## Inspect/Modify Dialog Box

The Inspect/Modify dialog box, illustrated below, allows you to examine RM/COBOL data items and to change the contents of an RM/COBOL data item.

To reveal or hide the Inspect/Modify dialog box, select a data item in a [Source Code/Edit window](#) (see page 97) and do one of the following:



- From the **View** menu, chose **Data Item**.
- Click the **View/Modify Data** toolbar button.
- Double-click the name of a data item.
- Right-click within the selection and choose **View Data Item** from the pop-up menu, or press **Shift+F9** or **Ctrl+I**.



Inspect/Modify Dialog Box	
Use this option	To
<b>Name</b>	Enter the name of the data item to be inspected or modified. You can enter a simple name or an expression as described in the “Data Address Development” topic of Chapter 9: <i>Debugging</i> of the <i>RM/COBOL User’s Guide</i> .
<b>Type</b>	<p>Select or enter the type of the data item. Specifying &lt;default&gt; displays the data item using the actual type of the data item. You can cause the value of a data item to be displayed as a different type by choosing a type from the drop-down list. See “Data Types” under the “General Debug Concepts” section in Chapter 9: <i>Debugging</i> of the <i>RM/COBOL User’s Guide</i>, for a description of allowed data types. In the second illustration above, the HEX data type was selected to show the binary data in INX-RECORD.</p> <p>If the data item is a POINTER data type (PTR), then selecting a non-PTR type from the Type list box causes CodeWatch automatically to dereference the pointer. (An indirect reference is sometimes called a dereferenced pointer.) To dereference the pointer, select the type of the pointed-to data from the list. The pointer will automatically be dereferenced to indicate that the displayed value was obtained indirectly. The value will be displayed in green. If the pointer has not been initialized, or the pointed-to memory cannot be accessed, an error message is displayed in red.</p> <p>To manufacture a pointer to a data item, select “&amp;PTR” from the Type list box. The value of a pointer to the specified data item will be displayed in green. (An “address of” modified reference is sometimes called a manufactured pointer.)</p> <p>One use for this function is to copy the value to the Windows Clipboard, and then replace the value of another pointer with the contents of the Clipboard. To copy and paste this value, select the text with the mouse and use the pop-up menu that appears when the right mouse button is clicked in the Show text box.</p>

Inspect/Modify Dialog Box	
Use this option	To
<b>Show (Button)</b>	Cause the current value of the data item to be displayed in the Show text box (the area below the Show button). If the value is available, that is, the data item is “in scope”, meaning the currently executing program includes or contains the data item, the value will be displayed in the Show box. If the data item is “out of scope”, or the value cannot be displayed using the specified type, a message is displayed in red.
<b>Show (Box)</b>	Display the current value of the data item. If the value is available, that is, the data item is “in scope”, meaning the currently executing program includes or contains the data item, the value will be displayed in the Show box. If the data item is “out of scope”, or the value cannot be displayed using the specified type, a message is displayed in red. Indirect references and addresses of references are displayed in green. See “Data Address Development” in Chapter 9: <i>Debugging of the RM/COBOL User’s Guide</i> , for more information on the pointer data item.
<b>Change</b>	Store any changes you made in the Show box into the data item, if possible. This button is disabled if the value is a constant item.
<b>Watch</b>	Add the data item to the list of watchpoints.

## Menus and Toolbars

The CodeWatch menu bar and toolbar are illustrated below. The menu bar appears at the top of the CodeWatch main window, immediately below the title bar, and is used to issue commands. The toolbars are, by default, arrayed below the menu bar. You can, however, drag a toolbar to a different docking location, or even float the toolbar free of the CodeWatch main window by holding down the Ctrl key while dragging.



**Note** The CodeWatch main window menu bar and toolbar vary depending on whether or not a program file has been opened and whether the program is running.

In the sections that follow, the toolbar buttons are illustrated with the corresponding menu commands. The CodeWatch menus include the following:

- [File menu](#) (see the following topic)
- [Edit menu](#) (see page 74)
- [View menu](#) (see page 75)
- [Build menu](#) (see page 76)
- [Run menu](#) (see page 77)
- [Window menu](#) (see page 79)

## File Menu

The File menu contains the following commands that operate on the current document, and on the workspace as a whole. The corresponding shortcut key and toolbar button, if applicable, are also shown.

File Menu			
Command	Shortcut Key	Toolbar	Action
<b>New</b>	Ctrl+N		<p>Creates a new, empty, unnamed source code document. The new document is not a file until it is saved for the first time, does not have a type, and is not part of the workspace.</p> <ul style="list-style-type: none"> <li>To give the new source code document a name, optionally give it a type (for example, COBOL source), and add it to the workspace by choosing the Save command on the File menu.</li> <li>Only files that are named and that are in the workspace may be compiled.</li> <li>Any number of source code documents may be opened at the same time.</li> </ul>
<b>Open</b>	Ctrl+O		<p>Opens a file. A dialog box that allows you to choose a file is displayed.</p> <p>This menu command can open source files, copy files, COBOL libraries, and COBOL program files, and automatically add them to the current workspace.</p> <p>If there is no current workspace, one will be created.</p> <p>You can select more than one file by holding down the Ctrl key and clicking on the name of the file with the mouse. However, all selected files will be added to the workspace as the same type.</p> <p>If a COBOL library is added to the workspace, a runtime /L Option naming the library will be passed to the runtime when execution begins. This is not the case for COBOL program files.</p>
<b>Close</b>			<p>Closes the current document.</p> <p>If the document contains unsaved changes, you will be asked if you would like to save the changes before the document is closed.</p> <p>If the document is maximized (that is, occupies the entire document area), the Close button is at the right end of the menu bar. Otherwise, the Close button is at the right end of the document title bar.</p>

File Menu			
Command	Shortcut Key	Toolbar	Action
Save	Ctrl+S		Saves the current document to a file. This command is available only if the document contains changes that have not been saved. If the document has not previously been saved into a file, then this command is the same as the Save As command.
Save As			Saves the current document to a specified file and then changes the name of the document to match the new filename. CodeWatch will ask you if you wish to overwrite an existing file. This command is not available for files that are part of the workspace. Use the Save Copy As command instead. This command is always available.
Save Copy As			Saves the current document to a specified file without changing the name of the document. CodeWatch will prompt if the specified file already exists. This command is always available.
Workspace   New	Ctrl+Shift+N		Creates a new workspace and displays the <a href="#">CodeWatch Workspace wizard</a> (see page 35). If a workspace is currently open, and the contents of that workspace have changed since the last save, you will be prompted to save or discard that workspace before the wizard is displayed. The Workspace wizard is displayed even if it has been disabled at startup.
Workspace   Open	Ctrl+Shift+O		Opens a previously saved workspace. If a workspace is currently open, and the contents of that workspace have changed since the last save, you will be prompted to save or discard that workspace before the new workspace is opened. The Open a Workspace File dialog box is displayed to allow you to choose the saved workspace, which is stored as a .cwf file.
Workspace   Close	Ctrl+Shift+F4		Closes the current workspace without creating or opening a new workspace. If the current workspace contains unsaved changes, you are asked if you want to save or discard the current workspace. If a program is running, it is automatically terminated when the workspace is closed. The Close Workspace button appears at the right end of the Workspace window title bar.

File Menu			
Command	Shortcut Key	Toolbar	Action
<b>Workspace   Save</b>	Ctrl+Shift+S		<p>Saves the current workspace to a file.</p> <p>This command is available only if the workspace contains changes that have not been saved.</p> <p>If the workspace has not previously been saved into a file, then this command is the same as <b>Workspace   Save As</b>.</p>
<b>Workspace   Save As</b>	Ctrl+Shift+A		<p>Saves the workspace to a specified file and then changes the name of the workspace to match the new filename.</p> <p>CodeWatch will ask you if you wish to overwrite an existing workspace file.</p> <p>This command is always available.</p>
<b>Workspace   Properties</b>	Ctrl+Shift+T		<p>Changes the workspaces properties. These options are discussed in <a href="#">Inspect/Modify dialog box</a> (on page 66).</p>
<b>Insert Items</b>			<p>Inserts one or more files into the workspace. The choices are:</p> <ul style="list-style-type: none"> <li>• COBOL Source File</li> <li>• COBOL Copy File</li> <li>• COBOL Library File</li> <li>• COBOL Program File</li> </ul> <p>Selecting an option displays the File Selection dialog box.</p> <p>This command is available only if there is a workspace.</p> <p>You can insert any number of files in one operation by holding down the Shift key when selecting files in the dialog box. All files must be the same type.</p> <p>The Insert Items command is also available on the pop-up context menu that appears when you right-click on an item or folder in the Workspace window. The item to be inserted is determined by the highlighted item or folder.</p> <p>The type of item added by the Ins key is also determined by the currently highlighted item in the Workspace window.</p>

File Menu			
Command	Shortcut Key	Toolbar	Action
<b>Remove Item</b>	Del		<p>Removes the item currently selected in the Workspace window from the workspace.</p> <p>Folders cannot be removed from the workspace.</p> <p>Source and copy files can be removed from the workspace only if they are not in use by a COBOL program or library. In this case, you must remove the program or library from the workspace.</p> <p>The Remove Item command is also available on the pop-up context menu that appears when you right-click on an item in the Workspace window.</p> <p>Note that the text of the menu item will change to reflect the action that will be performed.</p>
<b>Item Properties</b>	Ctrl+T		<p>Displays the properties of the selected item in the <a href="#">Item Properties dialog box</a> (see page 91).</p> <p>The tab that is displayed in the dialog box (Source File, Programs, or Build Options) is dependent on the item that is currently selected in the Workspace window.</p> <p>The item properties that can be examined include:</p> <ul style="list-style-type: none"> <li>• A list of included copy files.</li> <li>• A list of programs produced when the source file was compiled, or a list of source files that depend on the copy file.</li> <li>• For source files, the build settings that were in effect for the last compilation.</li> </ul> <p>The Build Options tab allows you to set the compilation options for the specified item, and permits you to set the default compilation options to be used for all files that do not have individual settings. See <a href="#">Setting Compilation Options</a> (on page 126) for more details.</p>

File Menu			
Command	Shortcut Key	Toolbar	Action
<b>Open Item</b>	F2		<p>Opens a Source Code/Edit window for the item that is currently highlighted in the Workspace window.</p> <p>The highlighted item may be:</p> <ul style="list-style-type: none"> <li>• A source or copy file. The specified file is opened.</li> <li>• A COBOL program. If the program was compiled with the /Y2 or /Y3 option, and the source code file is accessible, then the file is opened and the cursor is positioned to the beginning of the selected program.</li> <li>• A COBOL library or program file. If the library or program file contains programs that were compiled with the /Y2 or /Y3 option, and the source code file is accessible, then the file is opened and the cursor is positioned to the beginning of the selected program. If the library was created by the Combine Program utility (<b>rmppgmcom</b>) program, the source file for the first embedded COBOL program that was compiled with /Y2 or /Y3 is opened.</li> </ul> <p>Note that the text of the menu item will change to reflect the action that will be performed.</p>
<b>Edit Item with ...</b>	Alt+F2		<p>Invokes the external editor. If you have not configured an external editor, CodeWatch will issue a query as to whether you want to do so. If you do, CodeWatch displays the <a href="#">External Editing tab</a> (see page 110) of the Preferences dialog box so that you can configure the editor of your choice.</p> <p><b>Note</b> If an external editor has not been configured, the text of this command is Edit File "<i>filename</i>" with... .</p>
<b>Print</b>	Ctrl+P Shift+Ctrl+P		<p>Prints the selected source file. When selected, a sub-menu appears:</p> <ul style="list-style-type: none"> <li>• <b>Print:</b> displays the standard Windows Print dialog.</li> <li>• <b>Print Now!</b> Prints the current item on the default printer.</li> <li>• <b>Print Preview</b> opens the Print Preview window in the document area. The current item will be printed; however, if an editor selection is in effect, only the selected text will be printed.</li> <li>• <b>Print Setup</b> displays the Windows Print Setup dialog.</li> </ul>

File Menu			
Command	Shortcut Key	Toolbar	Action
<b>Recent Workspaces</b>			<p>Displays a list of recently opened workspaces. Choose one of the workspaces in the list to open the workspace quickly.</p> <p>You can quickly access a workspace in the list by pressing the key representing the digit in front of the workspace name. For example, to quickly reopen the most recently used workspace, press Alt+F to pull down the File menu, and then press the 1 key.</p>
<b>Exit</b>	Alt+F4		<p>Ends your CodeWatch session.</p> <p>If the current workspace has not been saved, you are asked either to save or discard the contents of the workspace.</p> <p>The Close button appears at the right end of the title bar in the CodeWatch main window.</p> <p>Any running program is automatically terminated before CodeWatch exits.</p>

## Edit Menu

The Edit menu contains the following commands that operate on the current selection or the object that currently has focus. The corresponding shortcut key and toolbar button, if applicable, are also shown.

Edit Menu			
Command	Shortcut Key	Toolbar	Action
<b>Undo</b>	Ctrl+Z or Alt+Backspace		<p>Undoes one level of changes to the current file. The change to be undone may be as simple as a typed character, or as complicated as a Replace that affected every line in the file.</p> <p>You can issue the Undo command repeatedly to undo additional changes.</p> <p>You can undo an undo operation with the Redo command (see below).</p>
<b>Redo</b>	Ctrl+Y or Ctrl+Backspace		<p>Reverses the effect of the last Undo operation. This command is available only after an Undo operation.</p> <p>An Undo command can be undone with a Redo command only if no additional changes have been made to the document since that Undo command was issued.</p> <p>If the Undo command has been issued repeatedly, the Redo command can also be used repeatedly to undo each level of Undo.</p>
<b>Cut</b>	Ctrl+X or Shift+Del		<p>Copies the current selection to the Windows Clipboard, and then deletes the selection.</p>

Edit Menu			
Command	Shortcut Key	Toolbar	Action
Copy	Ctrl+C or Ctrl+Ins		Copies the current selection to the Windows Clipboard.
Paste	Ctrl+V or Shift+Ins		Inserts the contents of the Windows Clipboard at the cursor position. If there is an active selection, the selected text is replaced with the text from the Windows Clipboard.
Select All	Ctrl+A		Selects all text in the current document.
Find	Ctrl+F or Alt+F3		Displays the <a href="#">Find dialog box</a> (see page 63), which allows you to search for text strings in the current document.
Find Next	F3		Searches for the next occurrence of the last text string for which you searched. The search is always toward the end of the file.
Find Previous	Shift+F3		Searches for the previous occurrence of the last text string for which you searched. The search is always toward the beginning of the file.
Replace	Ctrl+H		Displays the <a href="#">Replace dialog box</a> (see page 65), which allows you to search for and replace text strings in the current document.
Breakpoints	Ctrl+B		Displays the Code Breakpoints tab of the <a href="#">Breakpoints &amp; Watchpoints dialog box</a> (see page 87).
Watchpoints	Ctrl+W		Displays the Data Watchpoints tab of the <a href="#">Breakpoints &amp; Watchpoints dialog box</a> (see page 87).
Character Map	Ctrl+M		Displays the Character Map dialog box. This dialog box contains a grid that displays all the glyphs that are available in the currently selected font and character set (also called a “script”). This dialog box can be used to select a different font and character set. For a complete description, see <a href="#">Choosing an Editor Font</a> (on page 107). For further discussion, see also <a href="#">Character Sets</a> (on page 104).

## View Menu

The View menu contains the following commands. The corresponding shortcut key and toolbar button, if applicable, are also shown. When these commands are selected, a check mark (✓) appears to the left of the menu option.

View Menu			
Command	Shortcut Key	Toolbar	Action
Toolbar			Displays or hides the toolbars.
Status Bar			Displays or hides the status bar.

View Menu			
Command	Shortcut Key	Toolbar	Action
<b>Breakpoints Tab</b>	Alt+1		Displays the <a href="#">Breakpoints tab</a> (see page 52) in the Data window.
<b>Watchpoints Tab</b>	Alt+2		Displays the <a href="#">Watchpoints tab</a> (see page 53) in the Data window.
<b>Data Files Tab</b>	Alt+3		Displays the <a href="#">Data Files tab</a> (see page 54) in the Data window.
<b>Call Stack Tab</b>	Alt+4		Displays the <a href="#">Call Stack tab</a> (see page 55) in the Data window.
<b>Build Results Tab</b>	Alt+5		Displays the <a href="#">Build Results tab</a> (see page 56) in the Data window.
<b>Log Window</b>	Alt+6		Displays or hides the interactive debug <a href="#">Log window</a> (see page 58).
<b>Data Item</b>	Shift+F9 or Ctrl+I		Displays or hides the <a href="#">Inspect/Modify dialog box</a> (see page 66).
<b>Program Output</b>	F12		If the RM/COBOL program is currently running, reveals the RM/COBOL program output window and gives it the input focus.
<b>Preferences</b>	Alt+0		Displays the Preferences dialog box where you can customize the behavior of the Source Code/Edit window by using the <a href="#">Tabs tab</a> (see page 109) or the <a href="#">External Editing tab</a> (see page 110).

## Build Menu

The Build menu contains the following commands. The corresponding shortcut key and toolbar button, if applicable, are also shown.

Build Menu			
Command	Shortcut Key	Toolbar	Action
<b>Compile File</b>	Ctrl+F7		<p>Compiles the currently selected file or document.</p> <p>If a source file is opened in the Source Code/Edit window, the edit document containing the cursor is compiled.</p> <p>If the focus is in the Workspace window, and a COBOL source file is selected, that file is compiled. In this case, the menu item will contain the name of the file to compile.</p> <p>If the current source file has been edited and not saved, you will be asked to save the file or cancel the compilation.</p>

Build Menu			
Command	Shortcut Key	Toolbar	Action
<b>Compile Changes</b>	F7		<p>Compiles all files in the workspace that are marked as requiring recompilation.</p> <p>If a file to be compiled is open in a Source Code/Edit window, has been edited, and has not been saved, you will be asked to save the file or cancel the compilation.</p> <p>The following events will cause a file to be marked as needing recompilation:</p> <ul style="list-style-type: none"> <li>• If the file has been edited.</li> <li>• If a copy file used by the source file has been edited.</li> <li>• If the program file associated with the source file cannot be located, or if the timestamp of the source file is more recent than the timestamp of the program file.</li> <li>• If a compilation option has been changed either at the file or at the workspace level and compiling the file with the current options would affect the program.</li> </ul>
<b>Compile All</b>	Ctrl+Shift+F7		Compiles all source files in the workspace.
<b>Options</b>	Alt+F7		Displays the <a href="#">Build Options property tab</a> (see page 93) of the Item Properties dialog box for the currently selected source file. The options on this tab allows you to set compilation options for the current file, or for the entire workspace.
<b>Results   Find Next Results   Find Previous</b>	F4 Shift+F4		Displays a submenu where you can choose to move the cursor to the next or previous error or warning in the current file.

## Run Menu

The Run menu contains the following commands. The corresponding shortcut key and toolbar button, if applicable, are also shown.

Run Menu			
Command	Shortcut Key	Toolbar	Action
<b>Start Break Resume</b>	F5		Starts running the workspace. If the workspace program is already running, breaks execution at the next statement. Resumes running after a break. Note that the name of this command in the menu changes depending on the state you are in.
<b>Stop</b>			Stops the program. This is an orderly shutdown and all files will be properly closed. Execution can be restarted only from the beginning of the program. If the Persistent runtime attribute is set, the program cannot be restarted until you close the RM/COBOL program output window. See Chapter 3: <i>Installation and System Considerations for Microsoft Windows</i> of the <i>RM/COBOL User's Guide</i> .
<b>Step Statement</b>	F8		Executes one statement.
<b>Step Paragraph</b>	Shift+F8		Runs to the end of the current paragraph.
<b>Step Section</b>	Ctrl+F8		Runs to the end of the current section.
<b>Step Program</b>	Alt+F8		Runs until the current program returns or exits.
<b>Animate</b>			Enables or disables animation. When enabled, CodeWatch highlights each line as it is executed. When disabled, CodeWatch displays source code lines only when execution is paused or stops. When the program is running, you can pause execution at any time by pressing the F5 key in CodeWatch. When this command is selected, a check mark (✓) appears to the left of the menu option.
<b>To Cursor</b>	Shift+F5		Runs to the line in the current Source Code/Edit window that contains the cursor.
<b>To Program Change</b>	Alt+F5		Runs until the program highlighted by the cursor is entered or re-entered.
<b>Breakpoint</b>			Displays a submenu that allows you to set, clear, enable or disable, clear all, and show state of breakpoints. The commands on this submenu are described in the following table.
<b>Web Server Timers</b>			Only available when an Xcentricity Business Information Server (BIS) program is being run under CodeWatch. This option displays a submenu that allows BIS timers to be disabled to extend the debugging session, and can also be used to simulate a timer expiring.

The following functions are accessible from the **Run | Breakpoint** menu, the toolbar and the keyboard, and are enabled only while a source code file is open in the Source Code/Edit window.

Run   Breakpoint Menu			
Command	Shortcut Key	Toolbar	Action
Set			Inserts (or sets) a breakpoint at the line that contains the cursor.
Clear			Removes (or clears) the breakpoint at the current line.
Toggle Breakpoint	F9		Sets or clears a breakpoint at the current line. Note that the button appears pressed, and the menu option is checked if the current line contains a breakpoint.
Clear All Breakpoints			Removes (or clears) all breakpoints from the workspace.
State   Disable			Temporarily disables the breakpoint at the current line.
State   Enable			Enables the breakpoint at the current line.
State   Toggle	Ctrl+F9		Enables or disables a breakpoint at the line that contains the cursor. Note that the button appears pressed and the menu option is checked if the current line contains an enabled breakpoint.
State   Disable All			Disables all breakpoints in the workspace without removing the breakpoints.
State   Enable All			Enables all disabled breakpoints in the workspace.

## Window Menu

The Window menu contains the following commands. For additional information, see also [Using Document Windows](#) (on page 59).

Window Menu		
Command	Shortcut Key	Action
New Window	Ctrl+Z or Alt+Backspace	Opens a new document window that displays the same content as the current window. This command is available only when using a <a href="#">Source Code/Edit window</a> (see page 97).
Cascade		Cascades all document windows from left to right and top to bottom in the client area of the frame of the CodeWatch main window.
Tile		Arranges all document windows so they are visible in the client area of the frame of the CodeWatch main window.

Window Menu		
Command	Shortcut Key	Action
<b>Arrange Icons</b>		Arranges all minimized document windows across the bottom of the client area of the frame of the CodeWatch main window.
<b>Split</b>		Splits the current document window into two or four panes, each of which may be independently scrolled over the window content. This command is available only if the current window supports this operation. For more information, see <a href="#">Splitting Document Windows</a> (on page 61).
<b>Pane   Next Pane   Previous</b>	F6 Shift+F6	If the window has been split, moves the cursor into the next or previous pane in that window. For more information, see <a href="#">Cycling Through Open Windows</a> (on page 61).
<b>Docked</b>		If the current window is a dockable window, toggles the window between a docked window and a document window. When selected, the window is overlaid by two perpendicular intersecting lines, which may be positioned with the mouse or the keyboard. This command is available only if the current window supports this operation. For additional information, see <a href="#">Dockable Windows</a> (on page 46).
<b>Floating</b>		If the current window is a docked window, undocks the window and allow the window to float outside the frame of the CodeWatch main window. This command is available only if the current window supports this operation. For additional information, see <a href="#">Floating Windows</a> (on page 62).  Note that the window will automatically dock if moved over a docking site on the frame of the CodeWatch main window. To avoid docking, hold down the Ctrl key while moving the window.
<b>Open Windows</b>		Displays a list of currently opened document windows. To choose one of the windows in the list, type the number or click the name that corresponds to the Source Code/Edit window in which you want to work.

---

## Workspace Properties Dialog Box

The Workspace Properties dialog box contains the following tabs:

- [Program tab](#) (see page 81)
- [Options tab](#) (see page 83)
- [Environment tab](#) (see page 85)
- [Command tab](#) (see page 87)

These tabs correspond closely to the pages in the [CodeWatch Workspace wizard](#) (see page 35). The options available in the Workspace wizard pages are used to establish

properties when a workspace is being created, whereas the options available on the tabs in the Workspace Properties dialog box allow you to change the properties of an existing workspace. Each tab of the Workspace Properties dialog box is described in detail in the following sections.

To open the Workspace Properties dialog box, do one of the following:



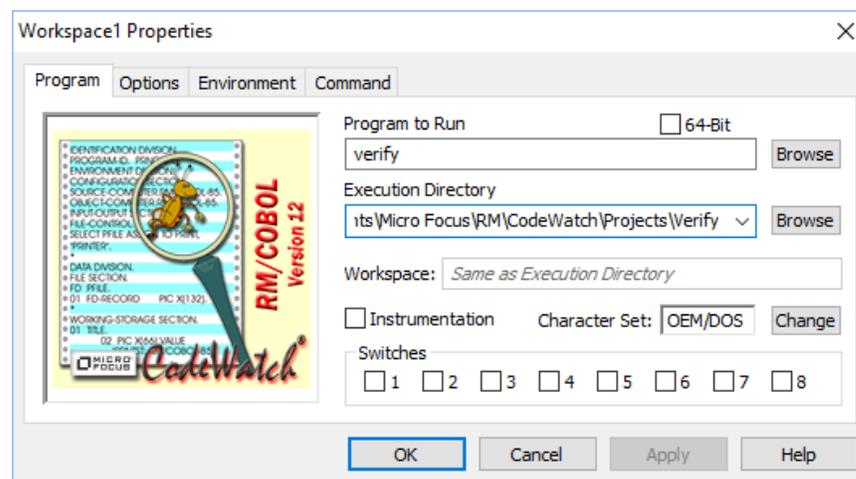
- From the File menu, choose **Workspace | Properties**.
- Click the **Properties** toolbar button.
- Press **Ctrl+Shift+T**.

Several buttons appear along the bottom of the Workspace Properties dialog box no matter which tab is selected and displayed. They perform the same function or action on all the tabs of the dialog box and are described below.

Workspace Properties Dialog Box	
Use this button	To
<b>OK</b>	Save changes and close the dialog box.
<b>Cancel</b>	Cancel the dialog box.
<b>Apply</b>	Apply changes without closing the dialog box. Changes made with the Apply button cannot be canceled by clicking the Cancel button.
<b>Help</b>	Display help for this tab of the dialog box.

## Program Tab

The Program tab of the Workspace Properties dialog box, illustrated below, allows you to specify the name of the main program to run, modify a character set for the workspace, and to define the execution environment for that program.



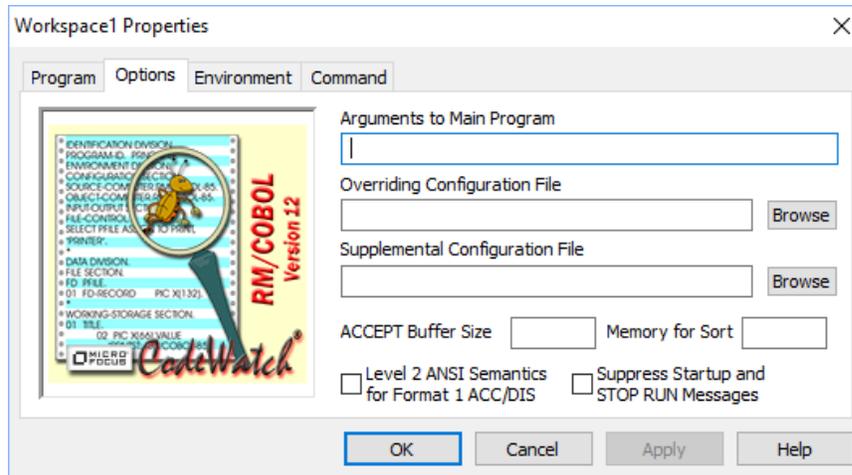
Program Tab of Workspace Properties	
Use this option	To
<b>Program to Run</b>	<p>Type or select the name of the COBOL program that CodeWatch is to run. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• The filename of a COBOL object file (for example, VERIFY.COB).</li> <li>• The name of a COBOL program to be found in a COBOL library (for example, VERIFY).</li> </ul> <p>CodeWatch uses the RM/COBOL runtime to locate programs (see the “Subprogram Loading” topic in Chapter 8: <i>RM/COBOL Features</i> of the <i>RM/COBOL User’s Guide</i>, for a complete description of the methods). For example, if VERIFY is specified as the program to run, and VERIFY is not found in any configured library, then the RM/COBOL runtime system will optionally search for a synonym for VERIFY in the environment, and will then search for a file named VERIFY.COB that contains exactly one separately compiled COBOL program in the current directory, the PATH, and the RUNPATH. CodeWatch does not attempt to resolve an ambiguous main program name until execution begins.</p> <p>This box may be left blank. If a main program is not specified, you will be asked for the program name when you run the workspace.</p> <p>Click the <b>Browse</b> button to open the Select a Program File dialog box where you can search for and select a COBOL program file (.cob).</p>
<b>64-Bit</b>	<p>When checked, the 64-bit runtime system will be used. The checkbox is disabled and unchecked if only a 32-bit runtime system is installed, and is checked and disabled if only a 64-bit runtime system is installed.</p>
<b>Execution Directory</b>	<p>Select the current directory that is used when the COBOL program begins execution. Click the down arrow in the list box to choose from a list of recently used working directories.</p> <p>Click the <b>Browse</b> button to open a Select a Directory dialog box where you can search for and select a working directory.</p>
<b>Workspace</b>	<p>This read-only field is the root location of the workspace. For new workspaces, this is initially the same location as the Execution Directory, but is set to the directory that contains the workspace file when the workspace is loaded or saved.</p> <p>Whenever possible, the files in the workspace are processed using this directory as a base directory. By saving the workspace into a directory that contains all of your source files, COPY files, and object files, those filenames can be stored as relative pathnames. This makes workspace files portable between machines, and makes it possible to check workspace .cwf files into version control systems..</p>
<b>Instrumentation</b>	<p>When selected, enable instrumentation. This option is equivalent to the /I Runtime Command Option.</p>

Program Tab of Workspace Properties	
Use this option	To
<b>Character Set</b>	<p>Show the character set for this workspace. The character set is used in the editor, during compilation, and during execution.</p> <p>Two character sets are allowed:</p> <ul style="list-style-type: none"> <li>• OEM/DOS</li> <li>• ANSI/Windows</li> </ul> <p>For a new workspace, the character set defaults to OEM/DOS after an installation of RM/COBOL. The RMSETNCS utility may be used to change the default character set for new workspaces to ANSI/Windows.</p> <p>After a workspace is created, the character set for that workspace remains the same unless explicitly changed by the user. The default character set for new workspaces can be overridden by specifying one of the <code>/cs_ansi</code> or <code>/cs_oem</code> command line options when starting CodeWatch.</p> <p>To change the character set, click the <b>Change</b> button, which is described in the next row.</p> <p><b>Note</b> For a detailed explanation of character sets within CodeWatch, see <a href="#">Character Sets</a> (on page 104). For further information about character sets and the RMSETNCS utility in RM/COBOL, see “Character Set Considerations for Windows” in Chapter 3: <i>Installation and System Considerations for Windows</i> of the <i>RM/COBOL User's Guide</i>.</p>
<b>Change</b>	<p>Open the <a href="#">Select Character Set dialog box</a> (see page 95) to change the character set assumed for the workspace. The dialog box also allows configuring the euro symbol for use in the CodeWatch editor.</p>
<b>Switches</b>	<p>Set or clear the current set of runtime switches. The eight switches are numbered consecutively from left to right. These options are equivalent to the <code>/S</code> Runtime Command Option.</p>

For a description of the buttons that appear along the bottom of this tab, see the [Workspace Properties Dialog Box](#) (on page 66).

## Options Tab

The Options tab of the Workspace Properties dialog box, illustrated below, allows you to specify additional execution parameters.



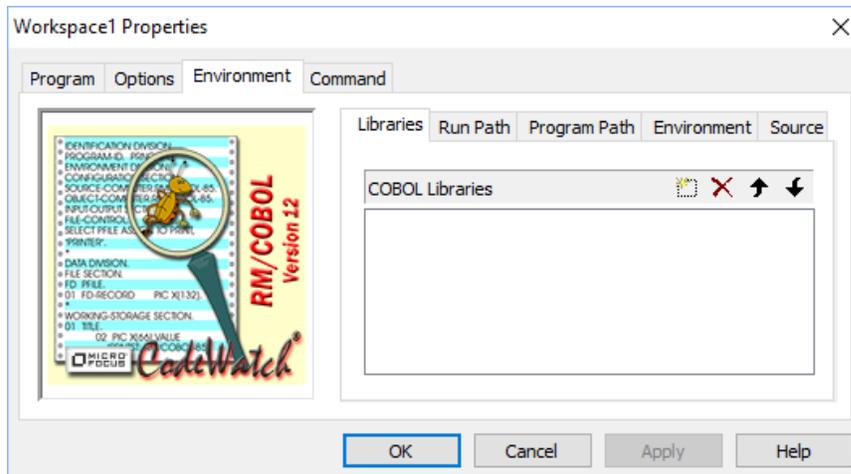
Options Tab of Workspace Properties	
Use this option	To
<b>Arguments to Main Program</b>	<p>Enter arguments to be passed to the main program. This option has the following format:</p> <p style="text-align: center;"><code>[delimiter]string[delimiter]</code></p> <p>The <i>delimiter</i> character may be either a single or double quote, must not appear within <i>string</i>, and is required only if <i>string</i> contains spaces or tabs. If you enter a string that contains spaces, but do not use a delimiter, CodeWatch will provide a delimiter or display an error if no delimiter can be chosen. This option is equivalent to the /A Runtime Command Option.</p>
<b>Overriding Configuration File</b>	<p>Select or enter the name of the primary runtime configuration file. This file completely overrides the default runtime configuration. This option is equivalent to the /C Runtime Command Option. If a relative path is specified, the path is interpreted as relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81).</p> <p>Click the <b>Browse</b> button to open the Select a Configuration File dialog box where you can search for and select a configuration file (.cfg).</p>
<b>Supplemental Configuration File</b>	<p>Select or enter the name of the supplemental runtime configuration file. The contents of this file supplement the default runtime configuration without replacing it. If an overriding configuration is specified, the contents of this file supplement that configuration. If no other configuration is specified, this file provides the complete configuration. This option is equivalent to the /X Runtime Command Option. If a relative path is specified, the path is interpreted as relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81).</p> <p>Click the <b>Browse</b> button to open the Select a Configuration File dialog box where you can search for and select a configuration file (.cfg).</p>
<b>ACCEPT Buffer Size</b>	<p>Enter the maximum buffer size to be used by ACCEPT and DISPLAY statements. The maximum size is 65280 characters, and the default size is 264 characters. This option is equivalent to the /B Runtime Command Option.</p>

Options Tab of Workspace Properties	
Use this option	To
<b>Memory for Sort</b>	Enter the amount of memory (in bytes) to be used for a sort operation. The default is 256000 bytes. Specify a larger value to improve the efficiency of sort operations. Decrease this value to allocate more memory for other data or called subprograms. If neither SORT nor MERGE operations will be used in this run unit, specifying 0 will free all reserved memory for other uses. This option is equivalent to the /T Runtime Command Option.
<b>Level 2 ANSI Semantics for Format 1 ACC/DIS</b>	When selected, indicate that Level 2 ANSI semantics are to be used for Format 1 ACCEPT and DISPLAY operations. The default is to use Level 1 ANSI semantics for these operations. This option is equivalent to the /M Runtime Command Option.
<b>Suppress Startup and STOP RUN Messages</b>	When selected, suppress the banner message and the STOP RUN message. This option is equivalent to the /K Runtime Command Option.

For a description of the buttons that appear along the bottom of this tab, see the [Workspace Properties Dialog Box](#) (on page 66).

## Environment Tab

The Environment tab of the Workspace Properties dialog box, illustrated below, allows you to specify the execution environment for your RM/COBOL program. The Environment tab contains five tabs. Click on a tab to set the parameters for the options available.



Environment Tab of Workspace Properties	
Use this option	To
<b>Libraries</b>	Specify a list of pathnames of COBOL and non-COBOL program libraries to be loaded. Libraries are loaded and searched in the specified order. The order can be changed by selecting a library and pressing the Move Up or Move Down buttons to move the name of the library within the list, or by dragging the name of the library to a new position with the mouse. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81). This option is equivalent to the /L Runtime Command Option.
<b>Run Path</b>	Specify a list of directories to be inserted before the RUNPATH environment variable. The RM/COBOL runtime will search these directories in the specified order for COBOL programs, libraries, non-COBOL libraries, and configuration files. These directories are inserted in front of any directories already in the RUNPATH environment variable. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81).
<b>Program Path</b>	Specify a list of directories to be inserted before the PATH environment variable. These directories are inserted in front of any directories already in the PATH environment variable. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81).
<b>Environment</b>	Specify a list of string pairs to be placed into the RM/COBOL program's environment. The syntax is: <i>name=value</i> Among other purposes, environment strings are used to specify synonyms for filenames. See the "Locating RM/COBOL Files on Windows" topic in Chapter 3: <i>Installation and System Considerations for Microsoft Windows</i> of the <i>RM/COBOL User's Guide</i> .
<b>Source</b>	Specify a list of directories to be searched for source files in the event that a source file cannot be found at the location specified in the object file. These directories are searched in the specified order, and the first file with a matching name is used. Relative pathnames are interpreted relative to the directory specified in the Working Directory option on the <a href="#">Program tab</a> (see page 81). For more information, see <a href="#">Locating Source Files During Execution</a> (on page 97) and the "Viewing Source During Program Execution" topic in <a href="#">Using the Source Code/Edit Window</a> (on page 101).

The following buttons appear on each of the tabs of the Environment tab and perform the same function on each tab.

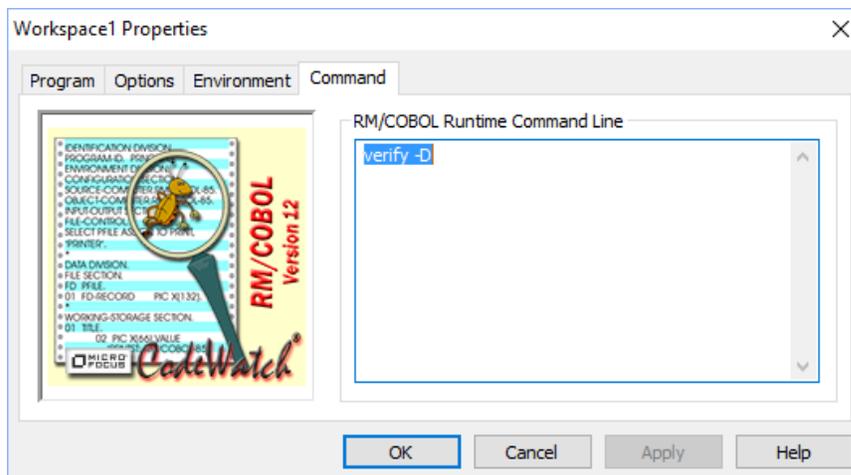
Environment Tab buttons	
Button	Name and Description
	<b>New.</b> Inserts a new item.
	<b>Delete.</b> Removes the selected item.
	<b>Move Up</b> (Alt+Up Arrow) and <b>Move Down</b> (Alt+Down Arrow). Moves the selected item up or down in the list.

Environment Tab buttons	
Button	Name and Description
	<b>Ellipsis.</b> On tabs that specify files or directories, clicking this button displays a File Open dialog box in order to search for and select a file or directory.

For a description of the buttons that appear along the bottom of this tab, see the [Workspace Properties Dialog box](#) (on page 66).

## Command Tab

The Command tab of the Workspace Properties dialog box, illustrated below, displays a text box where you can view or edit the RM/COBOL Runtime Command line options.



Command Tab of Workspace Properties	
Use this option	To
<b>RM/COBOL Runtime Command Line</b>	Display the command line that will be used to invoke the RM/COBOL runtime, based on the current configuration on the tabs in the Workspace Properties dialog box. Runtime Command line options may be changed by appending, deleting, or overtyping text in this text box. Any such changes will immediately be visible in the other properties tabs, but will not be applied to the workspace until either the OK or Apply button is pressed.

For a description of the buttons that appear along the bottom of this tab, see the [Workspace Dialog Properties Dialog Box](#) (on page 66).

## Breakpoints & Watchpoints Dialog Box

Breakpoints and watchpoints can be created, changed, and removed by using the Breakpoints & Watchpoints dialog box. This dialog box contains two tabs: Code Breakpoints and Data Watchpoints.

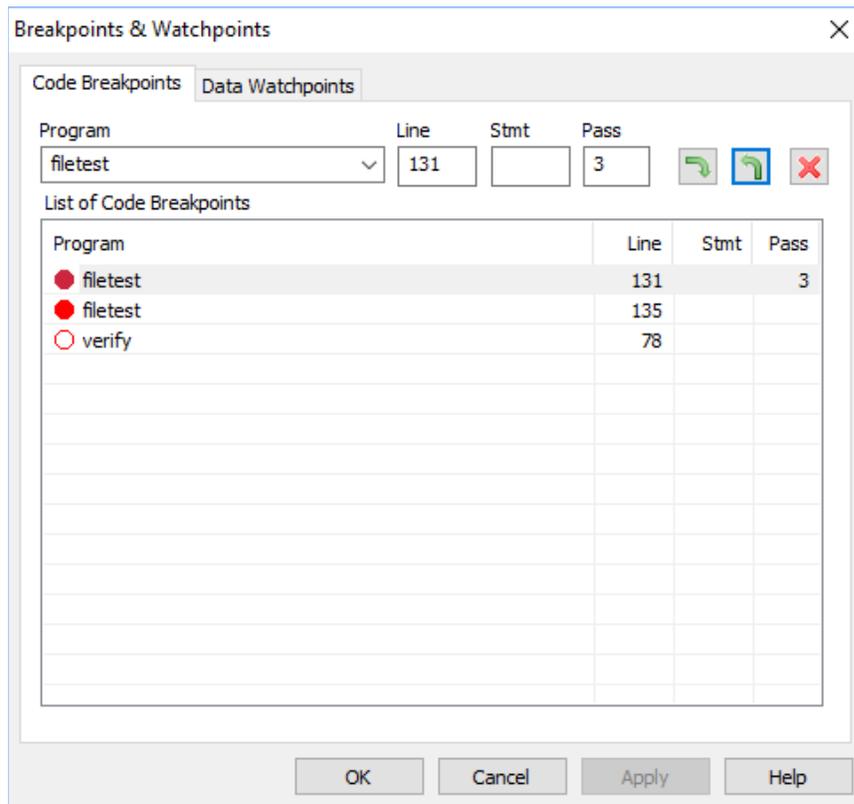
## Code Breakpoints Tab

The Code Breakpoints tab, illustrated below, allows you to create, remove, or change the properties of any or all breakpoints that are set in your workspace.

To display the Code Breakpoints tab, do one of the following:



- From the **Edit** menu, choose **Breakpoints**.
- Click the **Edit Breakpoints** toolbar button.
- Press Ctrl+B.
- From the [Breakpoints tab](#) (see page 52) of the Data window, right-click a program name and select **Properties** from the pop-up context menu or press **Alt+Enter**.



Code Breakpoints Tab	
Use this option	To
<b>Program</b>	Select or enter the name of the program that contains the breakpoint.
<b>Line</b>	Specify the number of the line that will cause a break to occur when executed.
<b>Stmt</b>	Specify the statement number within the line. Leave this blank to indicate that the break will occur when the first (or only) statement on the line is executed. Or, specify an offset: +1 indicates the second statement on the line, and +2 indicates the third.

Code Breakpoints Tab	
Use this option	To
<b>Pass</b>	Specify whether the breakpoint is to be skipped a number of times before being triggered. The default is to trigger the breakpoint on each pass. For example, a value of 5 indicates that the breakpoint will be skipped four times and then triggered. Once a breakpoint is triggered, it will be triggered on each subsequent pass unless the pass count is set again.
	Add or change a breakpoint in the list. If the program name, line number, and statement offset match another breakpoint in the list, that breakpoint is changed. Otherwise, a new breakpoint is added to the list.
	Copy the properties of a breakpoint into the text boxes.
	Remove the highlighted breakpoint from the list. The contents of the text boxes are ignored.
<b>List of Code Breakpoints</b>	Display a list of all code breakpoints in the current workspace. Each breakpoint is preceded by a red circle. A solid red circle indicates that the breakpoint is enabled; a hollow red circle indicates the breakpoint is disabled. This is followed by the name of the program, the line number, the statement number within the line, and the pass number.  A disabled breakpoint can be enabled, and an enabled breakpoint disabled by clicking on the red circle that precedes the breakpoint in this list.
<b>OK</b>	Apply all changes made to the list of breakpoints to the workspace and close the dialog box.
<b>Cancel</b>	Cancel the dialog box without saving changes.
<b>Apply</b>	Apply all changes made to the list of breakpoints to the workspace without closing the dialog box. Changes made with the Apply button cannot be canceled by clicking the Cancel button.
<b>Help</b>	Display help for this tab of the dialog box.

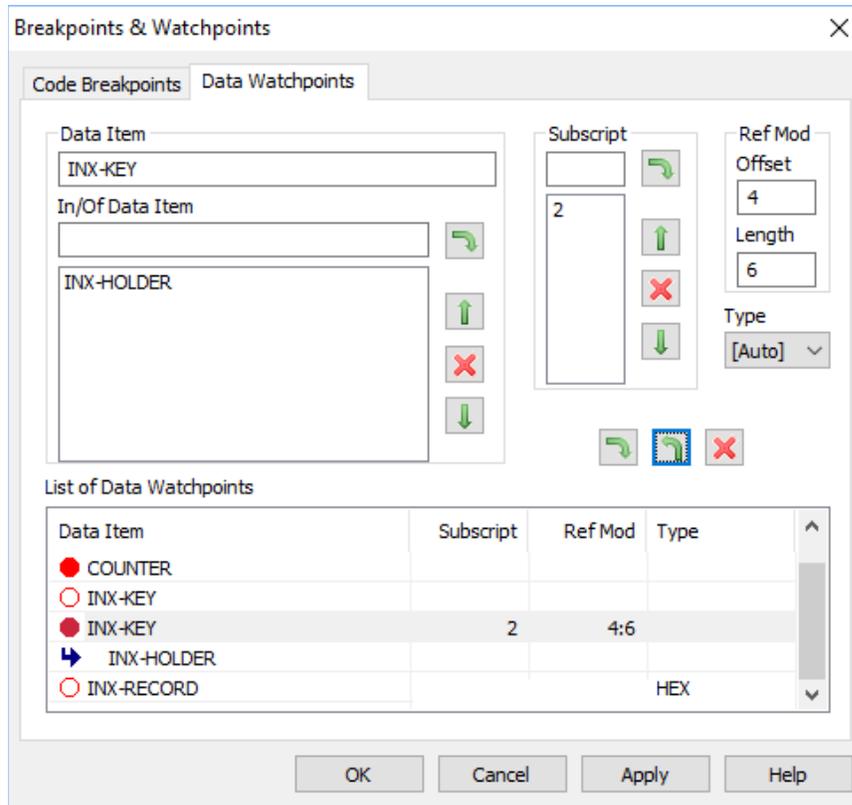
## Data Watchpoints Tab

The Data Watchpoints tab, illustrated below, allows you to create, remove, or change the properties of any or all watchpoints that are set in your workspace.

To display the Data Watchpoints tab, do one of the following:



- From the **Edit** menu, choose **Watchpoints**.
- Click the **Edit Watchpoints** toolbar button.
- Press **Ctrl+W**.
- From the [Watchpoints tab](#) (see page 53) of the Data window, right-click a data item name and select **Properties** from the pop-up context menu or press **Alt+Enter**.



Data Watchpoints Tab	
Use this option	To
<b>Data Item</b>	Enter the name of the data item being watched.
<b>In/Of Data Item</b>    	Specify qualification for the data item, if the name is not unique. The sequence of names must form a valid qualified reference to a data item, according to the rules for source program data references. After typing a data item name, click this button to add the item to the list. Use the up and down arrow buttons to arrange and order the data items in the list. Click this button to remove a data item from the list.
<b>Subscript</b>    	Specify a list of decimal integer subscripts that specify the table element to watch if Data Item is a table. After typing a decimal integer subscript, click this button to add the subscript to the list. Use the up and down arrow buttons to arrange and order the subscripts in the list. Click this button to remove a subscript from the list. The number of subscripts must match the number of OCCURS clauses in the hierarchy of data description entries for the data item. The value of each integer is interpreted as an occurrence number in the same way as a literal subscript in the source program.

Data Watchpoints Tab	
Use this option	To
<b>Ref Mod Offset and Length</b>	Optionally specify a subfield of the data item. Offset is a decimal number with a value ranging from 1 to the length of the data item. Length is a decimal number whose value ranges from 1 to the remaining length of the data item, after Offset has been applied. If Length is not specified, but Offset is specified, the length from offset to the end of the data item is presumed.
<b>Type</b>	Select the type of the watchpoint. This option is used to display the value of the data item in the <a href="#">Watchpoints tab</a> (see page 53) of the Data window. If the data item being watched is a PTR data item, it is possible to watch either the value of the PTR or the memory that the data item points to. To watch the pointer's value, select PTR from the Type list. To watch the pointed-to memory, select "type" from the list. For a description of allowed data types, see "Data Types" under the "General Debug Concepts" section in Chapter 9: <i>Debugging of the RM/COBOL User's Guide</i> .
	Add or change a watchpoint in the List of Data Watchpoints area.
	Copy the properties of a watchpoint into the text boxes.
	Remove the highlighted watchpoint from the List of Data Watchpoints area. The contents of the text boxes are ignored.
<b>List of Data Watchpoints</b>	Display a list of all data watchpoints in the current workspace. Each watchpoint is preceded by a red circle. A solid circle indicates that a trap on the watchpoint is enabled (a trap causes a break when the value of the watchpoint changes); a hollow circle indicates that the trap is disabled. This is followed by the name of the data item, the optional subscripts, the optional reference modifier, and the optional type.  A disabled trap can be enabled or an enabled trap disabled by double-clicking on the red circle preceding the name of the data item.
<b>OK</b>	Apply all changes made to the list of watchpoints to the workspace and close the dialog box.
<b>Cancel</b>	Cancel the dialog box without saving changes.
<b>Apply</b>	Apply all changes made to the list of watchpoints to the workspace without closing the dialog box. Changes made with the Apply button cannot be canceled by clicking the Cancel button.
<b>Help</b>	Display help for this tab of the dialog box.

## Item Properties Dialog Box

The Item Properties dialog box allows you to inspect and examine or change the properties of the currently selected item. This dialog box contains three tabs: Source File, Programs, and Build Options.

**Note** In this release of CodeWatch, only source files can have their properties inspected and/or changed.

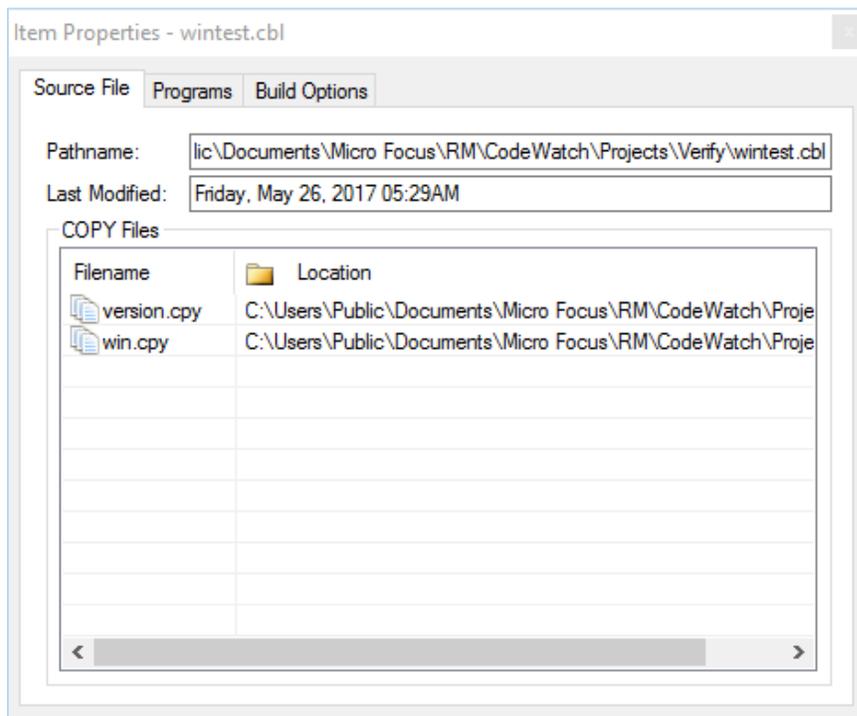
If a Source Code/Edit window is open, you can display the Item Properties dialog box by doing one of the following:

- From the **File** menu, choose **Item Properties** or click **Ctrl+T**.
- Select **Properties** from the workspace pop-up context menu that appears when you right-click in the Workspace window or Source Code/Edit window.
- Press **Alt+Enter** in the Source Code/Edit window or in the Workspace window.

**Note** To display the Item Properties dialog box by one of these methods when a Source Code/Edit window is not open, you must first select a source or copy file on the Files tab of the Workspace window.

## Source File Property Tab

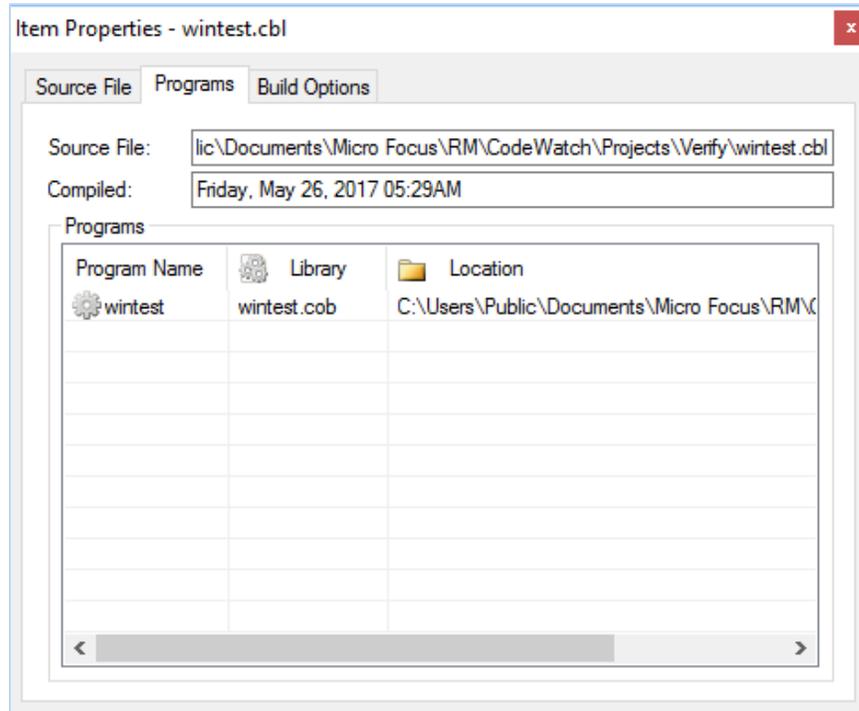
The Source File property tab, illustrated below, displays information about the currently selected source file.



Source File Property Tab	
Use this option	To
<b>Pathname</b>	Display the complete pathname of the currently selected source file.
<b>Last Modified</b>	Display the date/time that the source file was last changed. This is the same as the date of the file on the disk.
<b>COPY Files</b>	Display a list of copy files that were included when this source file was last compiled, including the name of the file and its location.

## Programs Property Tab

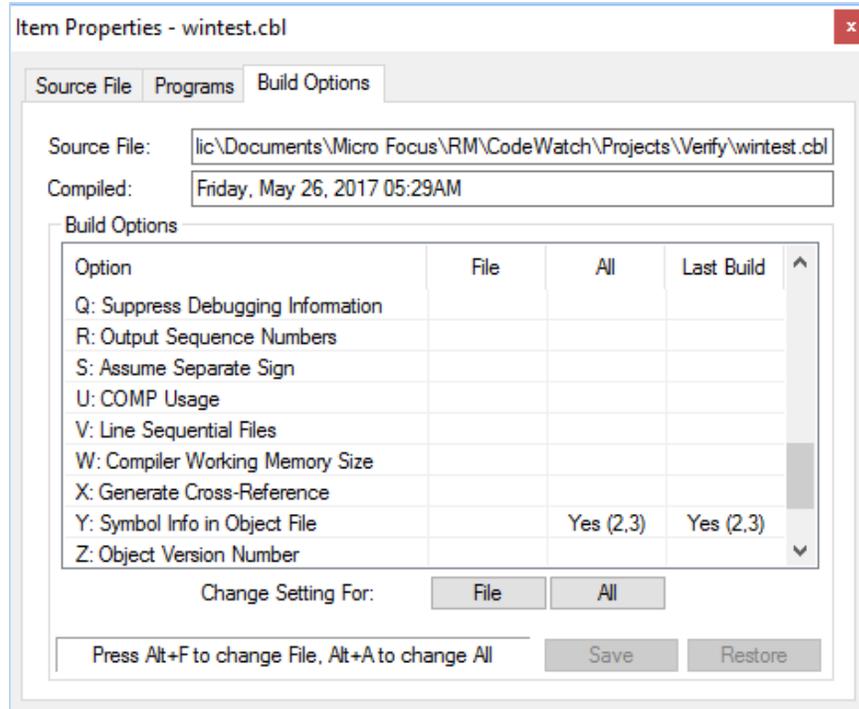
The Programs property tab, illustrated below, contains a list of programs that were generated when this source file was last compiled.



Program Properties Tab	
Use this option	To
<b>Source File</b>	Display the complete pathname of the currently selected source file.
<b>Compiled</b>	Display the date/time that the source file was last compiled.
<b>Programs</b>	Display a list of programs in the program file that were created when this file was compiled. The name of the program file, the containing library, and its location are available.

## Build Options Property Tab

The Build Options property tab, illustrated below, details the options that were in effect the last time this source file was compiled and the options that are currently in effect.



Build Options Property Tab	
Use this option	To
<b>Source File</b>	Display the complete pathname of the currently selected source file.
<b>Compiled</b>	Display the date/time that the source file was last compiled.
<b>Build Options</b>	<p>Display a list of the compilation options that were in effect the last time this source file was compiled. For each compilation option, this tab also specifies the default settings for the workspace and the settings for this file. The columns in the table provide the following information:</p> <ul style="list-style-type: none"> <li>• <b>Option.</b> A description of the RM/COBOL compilation option.</li> <li>• <b>File.</b> The setting for the currently selected file for each option.</li> <li>• <b>All.</b> The setting for the workspace. This setting is used if the File column is blank for a given option.</li> <li>• <b>Last Build.</b> The setting that was used when the file was last compiled. This setting is used if both the File and the All columns are blank for a given option.</li> </ul>
<b>Change Setting For</b>	<p>Change the setting in the column aligned with the button:</p> <ul style="list-style-type: none"> <li>• <b>File.</b> Cycles the setting for the currently selected option between the possible values for the option. You can press this button from the keyboard by typing Alt+F.</li> <li>• <b>All.</b> Cycles the setting for the currently selected option between the possible values for the option. You can press this button from the keyboard by typing Alt+A.</li> </ul> <p>For options that have discrete values (such as a file or directory name), a dialog box will appear to allow you to set the value.</p>

Build Options Property Tab	
Use this option	To
<b>Save</b>	Save the changes you made in this tab of the dialog box. Changes do not take effect until you click this button.
<b>Restore</b>	Restore this dialog box to its initial state, or to the point where you last clicked the Save button.

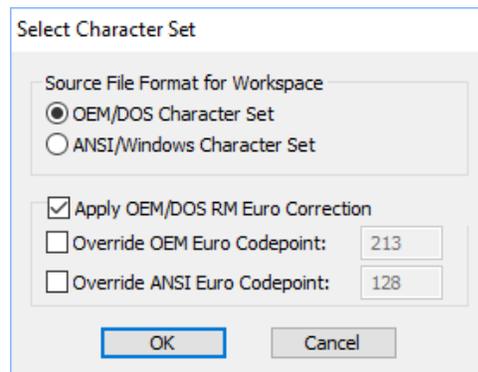
For more information about using the options on this tab, see [Setting Compilation Options](#) (on page 126).

## Select Character Set Dialog Box

The Select Character Set dialog box allows you to select the character set for a workspace and to configure the euro character used by the CodeWatch editor when an OEM/DOS character set is selected.

To display the Select Character Set dialog box, do one of the following:

- From the [Set Program to Run page](#) (see page 38) of the CodeWatch Workspace wizard, click the **Change** button.
- From the [Program tab](#) (see page 81) of the Workspace Properties tab, click the **Change** button.



Select Character Set Dialog Box	
Use this option	To
<b>OEM/DOS Character Set</b>	Select the OEM/DOS character set and enable the rest of the dialog box.
<b>ANSI/Windows Character Set</b>	Select the ANSI/Windows character set and disable the rest of the dialog box.
<b>Apply OEM/DOS RM Euro Correction</b>	Compensate for the fact that the OEM character set frequently does not contain a euro character and allow an OEM code point to be chosen that will be used to represent the euro. For OEM/DOS files, this code point is translated to the specified ANSI code point when the file is read into memory, and translated back when the file is written back to disk.

Select Character Set Dialog Box	
Use this option	To
<b>Override OEM Euro Code Point</b>	The code point value to use for the euro symbol in the OEM codepage. The value must be between 0 and 255. If the value is not specified, CodeWatch will query Windows for the correct code point to use. If the OEM codepage does not include a euro symbol (as is true for most OEM codepages other than 858), CodeWatch will use code point 213 (0xD5). This override is ignored if the Apply OEM/DOS RM Euro Correction checkbox is not selected.
<b>Override ANSI Euro Code Point</b>	The code point value to use for the euro symbol in the ANSI codepage. If the value is not specified, CodeWatch will query Windows for the correct code point to use. If the ANSI codepage does not include a euro symbol, CodeWatch will use code point 128 (0x80). This override is ignored if the Apply OEM/DOS RM Euro Correction checkbox is not selected.

The character set for the workspace can also be set by specifying `/CS_OEM` OR `/CS_ANSI` on the CodeWatch command line.

For further details on character sets in RM/COBOL for Windows, see “Character Set Considerations for Windows” in Chapter 3: *Installation and System Considerations for Windows* of the *RM/COBOL User's Guide*.

# Chapter 4: Source Code Editor

The internal CodeWatch Source Code Editor displays the source code for the program that is currently executing, if available, in the Source Code/Edit window. You can also edit the source code in the Source Code/Edit window.

This chapter discusses the following topics:

- [Locating Source Files During Execution](#) (see the following topic)
- [Contents of the Source Code/Edit window](#) (on page 98)
- [Using the Source Code/Edit Window](#) (on page 101)
- [Setting preferences](#) (on page 109)
- [Basic editing techniques](#) (on page 114)

---

## Locating Source Files During Execution

In order for CodeWatch to show the source code for a program during execution, the program must be compiled with RM/COBOL version 9 or later compiler, using the Full Debug Information (/Y2 or /Y3) Compile Command line option (see Chapter 6: *Compiling* of the *RM/COBOL User's Guide*). In addition, CodeWatch must be able to find the source file. CodeWatch looks for a source file by checking the following:

- The absolute path of the source file at the time it was compiled.
- The directory that contains the program file.
- The current working directory.
- The directories listed under the Source tab on the Environment tab of the [Inspect/Modify dialog box](#) (on page 66).

---

## Source Code/Edit Window

The Source Code/Edit window is opened when you perform the following operations:

- You use the **File | Open** command, the **File | New** command, or double-click on a program name, program file, library, source file, or copy file in the [Workspace window](#) (on page 47).

- The program is animating, a breakpoint or watchpoint is triggered, or the end of a step operation is reached.

A Source Code/Edit window does not close automatically. Other than memory, there is no limit on the number of Source Code/Edit windows that may be open at one time in a CodeWatch main window.

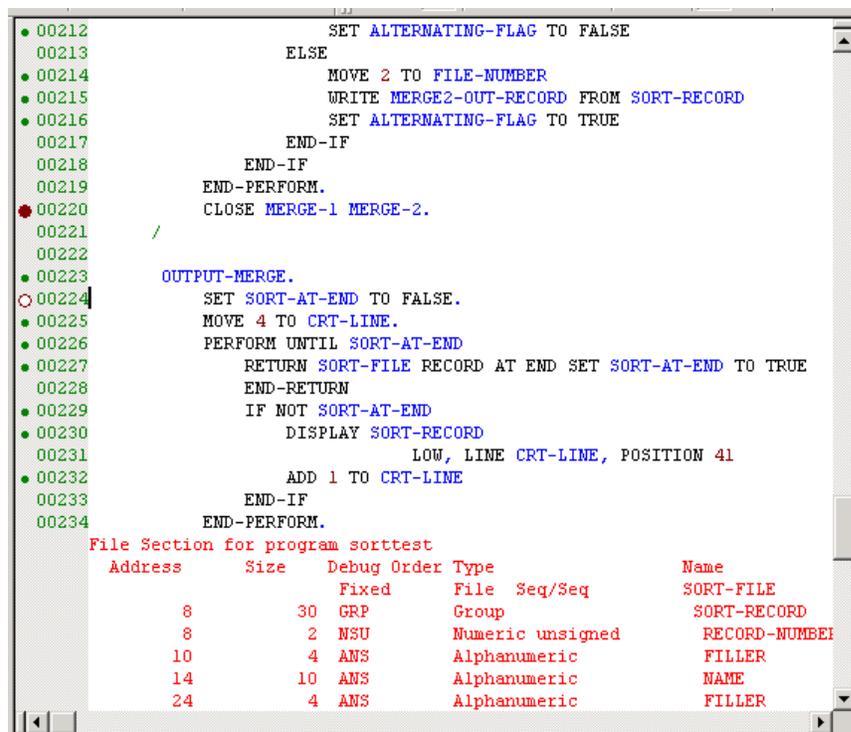
**Note** Windows 9x-class operating systems may have difficulties if more than 100 Source Code/Edit windows are open simultaneously. If performance decreases, close some of the windows. Windows NT-class operating systems will not experience this problem.

The following sections describe the elements of the Source Code/Edit window and the commands that are available from the context-sensitive menu that pops up when you right-click a selected item in this window.

## Contents of the Source Code/Edit Window

In the Source Code/Edit window, CodeWatch uses different colors to provide visual cues about the structure and state of your code.

The following illustrates a Source Code/Edit window.



```
00212      SET ALTERNATING-FLAG TO FALSE
00213      ELSE
00214      MOVE 2 TO FILE-NUMBER
00215      WRITE MERGE2-OUT-RECORD FROM SORT-RECORD
00216      SET ALTERNATING-FLAG TO TRUE
00217      END-IF
00218      END-IF
00219      END-PERFORM.
00220      CLOSE MERGE-1 MERGE-2.
00221      /
00222
00223      OUTPUT-MERGE.
00224      SET SORT-AT-END TO FALSE.
00225      MOVE 4 TO CRT-LINE.
00226      PERFORM UNTIL SORT-AT-END
00227      RETURN SORT-FILE RECORD AT END SET SORT-AT-END TO TRUE
00228      END-RETURN
00229      IF NOT SORT-AT-END
00230      DISPLAY SORT-RECORD
00231      LOW, LINE CRT-LINE, POSITION 41
00232      ADD 1 TO CRT-LINE
00233      END-IF
00234      END-PERFORM.
```

File Section for program sorttest

Address	Size	Debug Order	Type	Name
8	30	GRP	Fixed	SORT-FILE
8	2	NSU	Group	SORT-RECORD
10	4	ANS	Numeric unsigned	RECORD-NUMBER
14	10	ANS	Alphanumeric	FILLER
24	4	ANS	Alphanumeric	NAME
				FILLER

The main elements of the Source Code/Edit window are as follows:

- **Gutter.** This area is shown as lightly shaded with brown or gray and contains the following:
  - A green dot to indicate the line contains a statement.
  - A solid red circle to indicate that a breakpoint has been set on the current line.

- A hollow circle to indicate that a breakpoint has been set on the current line, and the breakpoint is temporarily disabled.
- A compiler-assigned line number. The line number is visible only if a program file has been found for this source file.
- **Text Area.** This is the area where source code is displayed and edited. Compiler diagnostic lines and lines included from copy files are also displayed in this area, and are displayed in red to indicate that they cannot be changed.

CodeWatch uses color to make your program easier to read:

- Keywords and reserved words are displayed in black.
- Paragraph and variable names are displayed in blue.
- Comments are displayed in green.
- Text that cannot be edited is displayed in red.
- Quoted text strings are displayed in cyan.
- Numbers are displayed in magenta.
- During execution, the active statement is highlighted in yellow, and a suspended statement (for example, one that issued a CALL) is highlighted in blue.

In addition to the items listed above, observe the following:

- **Statement Indicators.** A small diamond is displayed in the left margin to indicate that a line contains an executable statement. This is useful for setting breakpoints, because only breakpoints that are set on executable lines are honored. The indicators are displayed only in programs compiled with an RM/COBOL version 7.00.01 (or later) compiler. The indicators are not displayed when debugging programs compiled with the version 7.00.00 compiler; however, this does not affect debugging.
- **REPLACE Lines.** CodeWatch can only properly display listing lines that were modified by the REPLACE statement in programs compiled by the RM/COBOL version 7.00.03 (or later) compiler. If you are having trouble displaying source that uses REPLACE, recompile the program with the version 8.0 or later compiler. Note that COPY REPLACING operations are not affected.

**Note** The Source Code/Edit window is an example of a document window. For more in-depth information on how a document window works in CodeWatch, see [Document Windows](#) (on page 59).

## Context Menu for the Source Code/Edit Window

When you right-click in the Source Code/Edit window, a pop-up context menu appears. The contents of the context menu will vary depending on the current context.

Context Menu for the Source Code/Edit Window			
Command	Shortcut Key	Toolbar	Action
<b>Undo</b>	Ctrl+Z or Alt+Backspace		Undoes one level of changes to the current file: <ul style="list-style-type: none"> <li>• The change to be undone may be as simple as a typed character, or as complicated as a Replace operation that affected every line in the file.</li> <li>• You can issue the Undo command repeatedly to undo additional changes.</li> <li>• You can undo an Undo operation with the Redo command.</li> </ul>
<b>Redo</b>	Ctrl+Y or Ctrl+Backspace		Reverses the effect of the last Undo operation: <ul style="list-style-type: none"> <li>• This command is available only after an Undo operation.</li> <li>• An Undo operation can be undone only while no additional changes have been made to the document since the Undo command was issued.</li> <li>• If the Undo command has been issued repeatedly, the Redo command can also be used repeatedly to undo each level of Undo.</li> </ul>
<b>Cut</b>	Ctrl+X or Shift+Del		Copies the current selection to the Windows Clipboard and then deletes the selection.
<b>Copy</b>	Ctrl+C or Ctrl+Ins		Copies the current selection to the Windows Clipboard.
<b>Paste</b>	Ctrl+V or Shift+Ins		Inserts the contents of the Windows Clipboard at the cursor position. If there is an active selection, the selected text is replaced with the text from the Windows Clipboard.
<b>Select All</b>	Ctrl+A		Selects all text in the current document.
<b>Set Breakpoint</b> or <b>Remove Breakpoint</b>	F9		Toggles the breakpoint on the current line. Note that the text of this menu changes, and the button border indicates the presence or absence of a breakpoint on the current line.
<b>Enable Breakpoint</b> or <b>Disable Breakpoint</b>	Ctrl+F9		Enables or disables a breakpoint at the line that contains the cursor. Note that the menu option is checked, and the button border indicates the presence of an enabled breakpoint on the current line.
<b>View Data Item</b>	Shift+F9 or Ctrl+I		Displays or hides the <a href="#">Inspect/Modify dialog box</a> (on page 66) and initializes the data item field with the contents of the selection. If there is no selection, the word under the cursor is used.

Context Menu for the Source Code/Edit Window			
Command	Shortcut Key	Toolbar	Action
<b>Watch Data Item</b>	Alt+F9		Adds the current selection or the word under the cursor to the <a href="#">Watchpoints tab</a> (on page 53) of the Data window.  Note that this information will also appear in the List of Data Watchpoints area on the Data Watchpoints tab of the <a href="#">Breakpoints &amp; Watchpoints dialog box</a> (on page 87).
<b>Character Map</b>	Ctrl+M		Displays the Character Map dialog box. This dialog box contains a grid that displays all the glyphs that are available in the currently selected font and character set (also called a “script”).  This dialog box can be used to select a different font and character set. For a complete description, see <a href="#">Choosing an Editor Font</a> (on page 107). For further discussion, see also <a href="#">Character Sets</a> (on page 104).

## Using the Source Code/Edit Window

You can use the Source Code/Edit window to create new COBOL source files, open existing COBOL source files, view COBOL source files during program execution, and perform editing operations on source files.

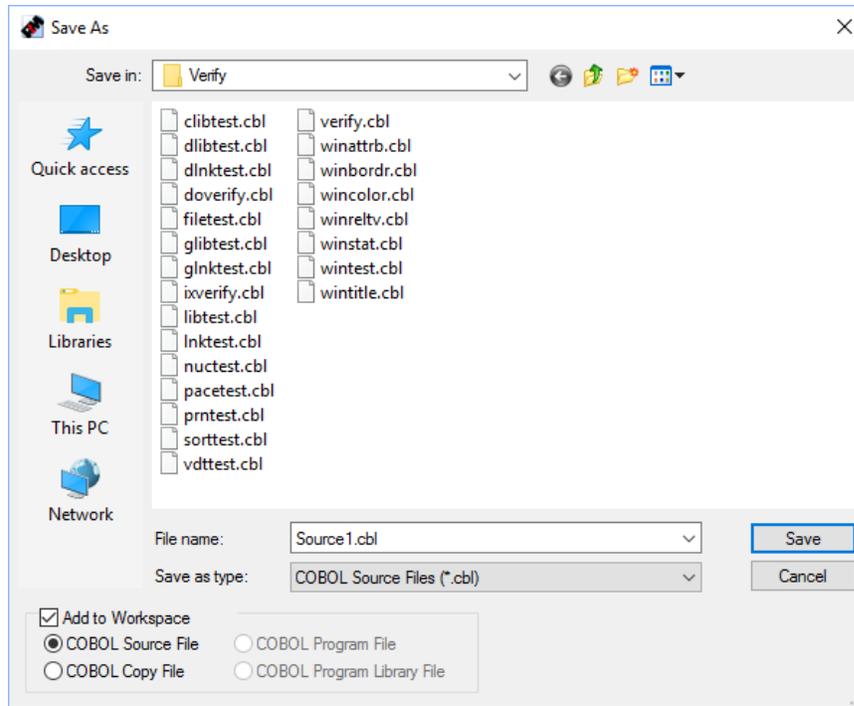
**Note** The Source Code/Edit window is an example of a document window. For more in-depth information on how a document window works in CodeWatch, see [Document Windows](#) (on page 59).

### Creating a New Source Code Document

To create a new, blank source code document:

- If a workspace has not been opened, from the **File** menu, choose **New | File**.
- Otherwise, use the **File | New** command.

A blank Source Code/Edit window will appear, containing an untitled document. The document is not part of the workspace and is therefore not a COBOL source file until the file is saved for the first time. At that point, the following dialog box is displayed:



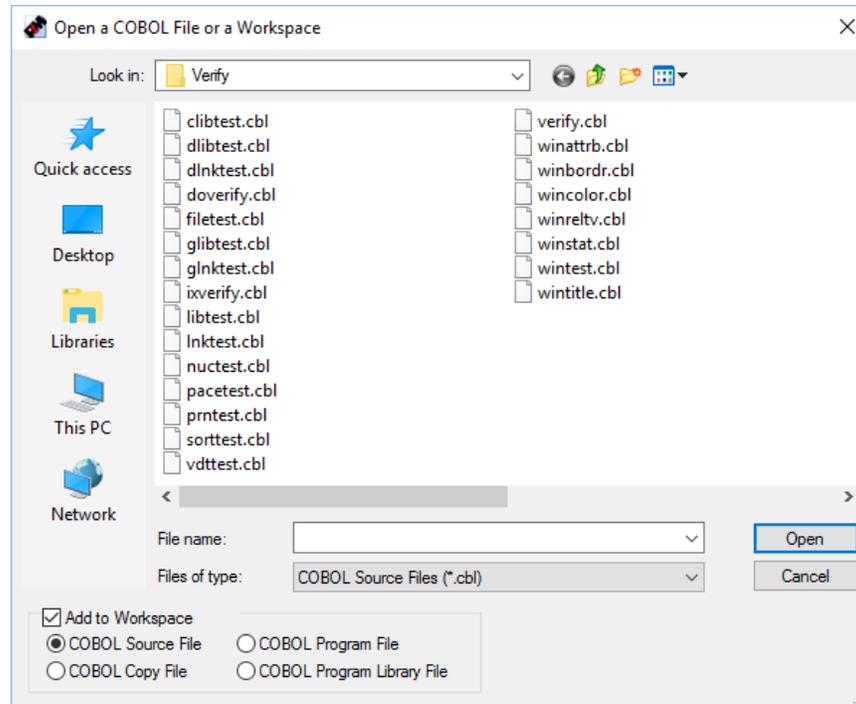
The file can be given any valid name—it does not need to have a **.cbl** extension to be treated as a COBOL source file. Instead, make sure **Add to Workspace** is checked and select either **COBOL Source File** or **COBOL Copy File**, as required, and click **Save**.

**Note** Only COBOL source files that are part of the workspace may be compiled from within CodeWatch.

## Opening an Existing Source Code File

To open an existing source code file, do one of the following:

- Use the **File | Open** command. This command displays the following Open File dialog box:



Select a file from the list. Check **Add to Workspace** if the file is a COBOL source or copy file and you would like to include the file in the workspace—in this case, be sure to also specify the kind of file you are opening.

If a workspace is not open, the **File | Open** command defaults to opening a workspace. To open a source file, pull down the **Files of Type** combo box and choose any option except **CodeWatch Workspace Files**. The **Add to Workspace** options will be enabled. Then, choose your file and select the proper type. A default workspace will be created containing the selected file.

- Select an item in the Workspace window and select the **File |Open Item** command. You can open the source for the selected item (which may itself be a source file) by right-clicking on the item and choosing **Open Item** from the menu, or by selecting the item and pressing the **F2** (or open) key.

## Viewing Source During Program Execution

The Source Code Editor will automatically display your source code during program execution as long as the program was compiled with the /Y2 or /Y3 option, and the source for the program is accessible. The program file, program, and attendant source and copy files are also automatically inserted into the workspace when opened in this way.

## Editing Operations in the Source Code/Edit Window

You can perform the following editing operations in a Source Code/Edit window:

- Enter new text by clicking with the mouse to set the insertion point, and then typing.
- Select regions of text with the keyboard or the mouse.

- Delete a selected region of text, or copy or cut a region to the Windows Clipboard.
- Paste text from the Windows Clipboard at the insertion point.
- Right-click to display a pop-up menu.
- Undo or redo one or more operations.

To learn more, see [Basic Editing Techniques](#) (on page 114).

You can also perform the following non-editing operations in a Source Code/Edit window:

- Set a new breakpoint or clear an existing breakpoint by double-clicking in the indicator column or on the line number. You can set, clear, enable, or disable a breakpoint by right-clicking on a line and choosing the appropriate item from the pop-up context menu, or by using buttons on the toolbar.

For a breakpoint to be effective, it must be set on a line that contains executable code. Such lines are indicated by a small green dot in the gutter. Comment lines are not executed; neither are the second or subsequent lines of statements that span more than one line.

- Examine a data item during execution. To do this, highlight the data item in the Source Code/Edit window by clicking with the mouse; then, double-click the selected item to display the [Inspect/Modify dialog box](#) (on page 66). If the data item requires a subscript or qualification, you can edit the name in the Inspect/Modify dialog box. The value of the data item will be updated the next time the Debugger reaches a breakpoint or watchpoint, or you can force the value to be refreshed immediately by choosing **Break** from the **Run** menu, or by pressing F5. You can also add a watchpoint by clicking the **Watch** button, or modify the data item by entering a new value and clicking the **Change** button in this dialog box.

You can also set a watchpoint by right-clicking on a highlighted data item with the mouse and choosing **Watch Data Item** or by pressing **Alt+F9** from the pop-up menu. Note that you cannot specify qualification or subscripts in this way, but you can edit the watchpoint later to add this information.

- Search for a text string by doing one of the following:
  - From the **Edit** menu, choose **Find**.
  - Click the **Find** toolbar button.
  - Press **Alt+F**.



You can search again by using the Edit menu and choosing the **Find Next** or **Find Previous** command (or by pressing the corresponding toolbar buttons). As an alternative, you can also use the shortcut keys. Press **F3** to search forward or press **Shift+F3** to search backward in the file.

---

## Character Sets

A codepage defines a mapping of character code points (often called bytes) to a set of characters. CodeWatch initially uses the default character set contained in the Courier New font to display and edit text. For systems installed in the United States and Western Europe where the default codepage is 1252, this will be the ANSI character set (also referred to as “Western” script or the Latin-1 code set). Systems installed in other locales will use the default codepage for that locale.

However, many RM/COBOL programs were originally written for and executed under DOS. The codepage/character set used by DOS programs is hardware-specific, and is frequently referred to as the OEM/DOS<sup>1</sup> character set.

The difference between the ANSI and OEM character sets is not important if the COBOL program uses only the 7-bit characters in the ASCII character set. The lower half of all Windows-supported ANSI and OEM codepages, code points 000 – 127 (0x00 – 0x7F), always match because they represent the same ASCII character set. The glyph (the visual representation of a character) for these code points is the same in both the ANSI and OEM character sets (and most others as well).

However, this is not true for code points above this range. The upper half of Windows ANSI and OEM codepages, code points 128 – 255 (0x80 – 0xFF) can differ significantly in the characters that particular code points represent. For example, [Figure 1](#) illustrates the glyphs displayed for code points 128 to 255 decimal when the Western character set in the Courier New font is selected.

**Figure 1: Glyphs in Western (ANSI) Character Set in “Courier New” Font**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	€	□	ı	f	„	…	†	‡	^	‰	Š	<	Ⓒ	□	Ž	□
90	□	˘	ı	ˆ	ˆ	•	—	—	˜	™	Š	>	œ	□	ž	Ÿ
A0	□	ı	ç	£	¤	¥	ı	š	ˆ	©	ª	«	¬	–	®	—
B0	°	±	²	³	´	µ	¶	·	,	±	°	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

[Figure 2](#) illustrates the glyphs displayed for code points 128 to 255 decimal (or 80 through FF hex) when the OEM/DOS character set in the Terminal font is selected for an OEM codepage 437.

**Figure 2: Glyphs in OEM Character Set in “Terminal” Font**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	Ç	Ü	é	â	ä	à	á	ç	ê	ë	è	ì	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	ú	û	ÿ	Ö	Ü	¢	£	¥	₣	ƒ
A0	á	í	ó	ú	ñ	Ñ	®	©	¿	¬	¬	½	¼	ı	<<	>>
B0	▒	▒	▒					π	ƒ			π	π	π	π	π
C0	L	⊥	⊥	⊥	—	+	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	Π	Σ	σ	μ	γ	ϕ	θ	Ω	δ	∞	ø	€	∩
F0	≡	±	≥	≤	↑	J	÷	≈	°	•	•	√	ˆ	²	•	□

In particular, note the highlighted first glyph in the first column (code point hex 80 or decimal 128). If a Western (ANSI) character set is being used to display this code point, it will appear as a Euro glyph. However, if an OEM/DOS character set is being used to display this code point, it will appear as a C-cedilla glyph. The combination of font and character set is critical if non-ASCII text is to be displayed correctly.

**Note** While most Windows fonts contain more than one character set, to ensure that characters entered into COBOL literals can be displayed by the RM/COBOL runtime, CodeWatch allows only the default character set in each font to be selected for text editing. An exception is made when using the OEM character set for a workspace; in this case, the OEM/DOS script for a font, if it exists, may be selected.

## The Workspace Character Set

Because RM/COBOL source files are simply text files, it is not possible for CodeWatch to determine automatically the character set that was used to create the file. Instead, each CodeWatch workspace has an implicit character set that can be set in three ways:

- From the command line when CodeWatch is started. See [System Installation](#) (on page 8).
- With the Workspace wizard when the workspace is created. See the Character Set and Change options on the [Set Program to Run page](#) (on page 38).
- With the **Character Set** option on the [Program tab](#) (see page 81) in the Workspace Properties dialog box after the workspace has been created.

If the source and copy files use only code points in the ASCII character set, that is, all code points are in the range of 32 to 127 decimal, no further action is required. These code points refer to the same characters in both ANSI and OEM codepages.

For code points outside of this range, there are two options:

1. If you know that the source files were created using the ANSI character set, you simply have to choose a font that contains the correct character set for the current locale. The default is normally just fine. Note that for ANSI workspaces, you must choose a font that defaults to an ANSI character set (this includes most Windows fonts). You cannot choose a font that defaults to the OEM character set. It is not required that the default character set be “Western”—it can be any non-OEM character set.
2. If the source files were created with the OEM/DOS character set (which is often the case for legacy projects originally developed for DOS), you should declare that the workspace uses OEM/DOS files. In this instance, CodeWatch will translate the characters from OEM/DOS to ANSI when they are loaded into the editor, and will translate them back from ANSI to OEM/DOS when the file is rewritten. This translation process is bi-directional: if you choose an ANSI font, not all OEM glyphs will display correctly (ANSI fonts do not contain line-drawing glyphs), but the glyphs will be preserved and not altered unless you choose to overtype them.

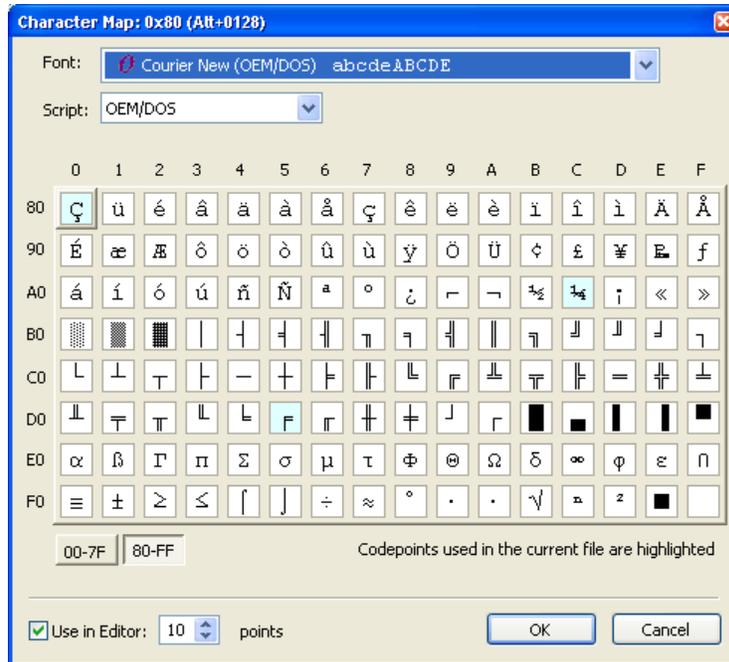
To ensure that all OEM/DOS characters are visible in the editor, you should choose to edit with a font that contains an OEM character set. CodeWatch will normally allow you to edit only with the default character set in a font, but this rule is relaxed for OEM character sets—the Character Map dialog box allows you to choose any font that contains an OEM character set. Please be aware that not all these fonts are available to running COBOL programs.

The next section describes how to choose a font.

## Choosing an Editor Font

The CodeWatch Character Map dialog box allows you to choose a font for the current mode. CodeWatch tracks fonts separately for ANSI and for OEM workspaces. To open the Character Map dialog box, illustrated below, choose the **Character Map** command either from the [Edit menu](#) (see page 74) in the CodeWatch main window or the [Context menu for the Source Code/Edit window](#) (see page 99).

**Note** This menu command is available only when a file is being edited in the Source Code/Edit window.



The Character Map dialog box has the following options:

Character Map Dialog Box	
Use this option	To
<b>Font</b>	<p>Choose a font. The list of available fonts is governed by three variables: the workspace character set, the selected script (see "Script" in the following row), and the <b>Fixed Pitch Only</b> check box.</p> <p>The font name is prefixed with an O symbol if the font is an OpenType font or a TT symbol if the font is a TrueType font. No prefix indicates a raster font, which is less desirable.</p> <p>Each font is suffixed with the selected character set and a short sample.</p>

<b>Character Map Dialog Box</b>	
<b>Use this option</b>	<b>To</b>
<b>Script</b>	<p>Restrict the list of fonts to those that contain a specific character script or character set.</p> <p>On Windows, fonts can contain multiple character sets, but not all fonts contain every character set. CodeWatch offers a set of fonts and character sets appropriate for either the ANSI or the OEM mode of the workspace. The <b>Script</b> dropdown list allows you to see all valid fonts, or just those that contain a particular script.</p> <ul style="list-style-type: none"> <li>• If the workspace mode is OEM, you will see a list of all fonts that use OEM as the primary character set, those fonts that provide an OEM character set (even if it is not the primary character set), and those fonts that use ANSI as the primary character set.</li> <li>• If the workspace is ANSI, you will see a list of all fonts that use a non-OEM character set as the primary character set. You will not see OEM fonts.</li> </ul> <p>Selecting <b>All</b> from this box lists all the fonts that may be used in the workspace.</p>
<b>Character Grid</b>	<p>Display the character glyphs for each code point in the selected font and character set. This is the font and character set being used in the editor only if the <b>Use in Editor</b> check box is selected.</p> <p>By default, the code points that have ordinal values between 128 and 255 (or 80 and FF hex) inclusive are displayed. To display code points between 0 and 127 (or 0 and 7F), inclusive, click the appropriate button at the bottom of the grid.</p> <p>Code points that are actually being used in the file that is being edited are highlighted.</p> <p>To see the code point of a particular glyph, hover over the code point with the mouse. The title bar of the dialog box will display the hexadecimal value of the code point and the Alt key combination that will insert this code point into the text editor. For example, using the Courier New font and the OEM/DOS character set (as in the example above), hover over the degree symbol in row A0 and column 7. The title bar will change to the following:</p> <p style="text-align: center;">Character Map: 0xA7 (Alt+0167)</p> <p>To insert this character in the editor, close the Character Map dialog box to return to the editor. Then hold down the Alt key and press 0167 on the numeric keypad. Finally, release the Alt key. The degree symbol ° will appear at the cursor position.</p> <p>Note that the code point of each symbol may vary depending on the character set.</p>
<b>Use in Editor</b>	<p>Use the currently selected Font and Size combination in the editor. When the Character Map dialog box is first displayed, the currently selected font and size are displayed and the check box is selected. The box is automatically unchecked if a different font or script is selected, but not if a different font size is selected. If this check box is selected when you press the OK button, the currently selected font and size will be used for editing.</p>
<b>OK</b>	<p>Click this button to optionally change the font face name and size in all open editors. Note that the change is applied only if the <b>Use in Editor</b> check box is selected.</p>
<b>Cancel</b>	<p>Close the dialog box without changing the font face name and font size in the editor.</p>

## Notes

- Only fixed-pitch fonts can be used in the editor.
- Changing the font face name and font size affects all open editors immediately.
- External editors are not affected by any of the discussion regarding [character sets](#) (see page 104). For a workspace with an OEM character set, only an external editor that uses the OEM codepage should be used. For a workspace that uses the ANSI codepage, only an external editor that uses the ANSI codepage should be used. For information on setting an external editor, see [External Editing tab](#) (on page 110).

---

# Setting Preferences

The behavior of the Source Code Editor can be customized and controlled by using the Preferences dialog box.

To display the Preferences dialog box:

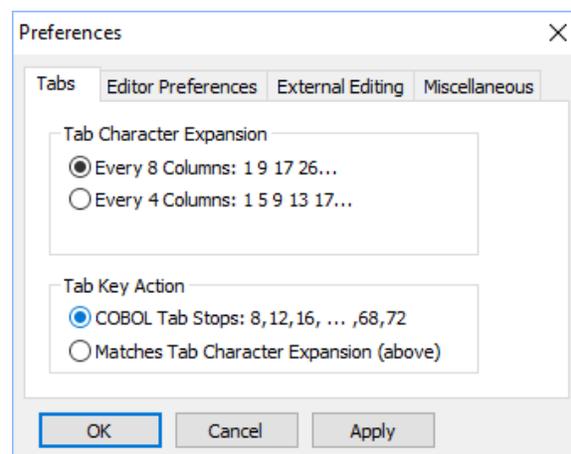
- From the **View** menu, choose **Preferences** or press **Alt+0**.

This dialog box contains the following tabs:

- [Tabs tab](#) (see page 109)
- [External Editing tab](#) (see page 110)

## Tabs Tab

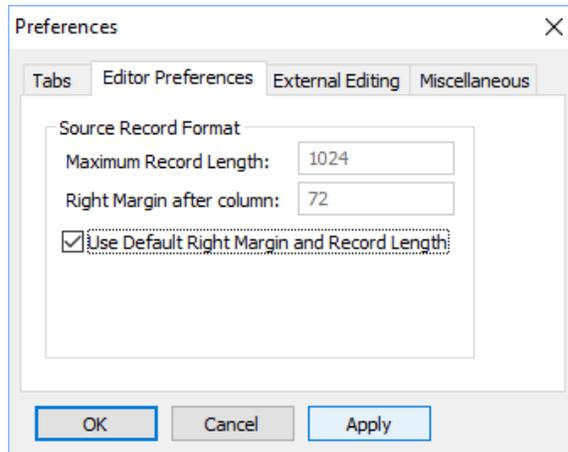
The Tabs tab, illustrated below, allows you to specify the way CodeWatch expands tab characters when displaying a file in the Source Code/Edit window and determines how the CodeWatch Source Code Editor behaves when the Tab key is pressed on the keyboard in Insert mode.



<b>Tabs Tab</b>	
<b>Use this option</b>	<b>To</b>
<b>Tab Character Expansion</b>	<p>Select how CodeWatch expands tab characters when displaying a file in the Source Code/Edit window.</p> <ul style="list-style-type: none"> <li>• <b>Every 8 Columns.</b> Selecting this option indicates that a tab character causes the cursor to advance by 1 to 8 spaces so the next character is displayed on an 8-character boundary. This is the standard interpretation of tabs by DOS and by many UNIX systems and is the default setting.</li> <li>• <b>Every 4 Columns.</b> Selecting this option indicates that a tab character causes the cursor to advance by 1 to 4 spaces so the next character is displayed on a 4-character boundary.</li> </ul> <p><b>Notes</b></p> <p>The Tab Character Expansion setting applies to all Source Code/Edit windows.</p> <p>CodeWatch does not insert tab characters; this setting controls how existing tab characters in files are interpreted. If you edit a line, CodeWatch will replace tabs with spaces.</p> <p>When compiling in CodeWatch, the compiler expands tabs the same way that CodeWatch expands tabs. This behavior can be overridden by specifying a compiler configuration file on the <a href="#">Build Options property tab</a> (see page 93); however, overriding tab settings in this way is not recommended.</p>
<b>Tab Key Action</b>	<p>Determine how the CodeWatch Source Code Editor behaves when the Tab key is pressed on the keyboard in Insert mode.</p> <ul style="list-style-type: none"> <li>• <b>COBOL Tab Stops.</b> Selecting this option indicates that the cursor is advanced to the next COBOL tab position. If the cursor is in columns 1-7, the cursor is advanced to column 8 (area A). Otherwise, the cursor is advanced to the next 4-character boundary. This is desirable behavior when coding COBOL and is the default setting.</li> <li>• <b>Matches the setting above.</b> Selecting this option indicates that the cursor is advanced either to the next 8-character, 4-character, or COBOL tab boundary, as specified by the settings in the Tab Character Expansion area.</li> </ul> <p><b>Notes</b></p> <p>The Tab Key Action setting applies to all Source Code/Edit windows.</p> <p>CodeWatch does not insert tab characters. This setting controls how the cursor moves, and how many spaces are inserted when the Tab key is pressed.</p>
<b>OK</b>	Apply any changes made above and close this dialog box.
<b>Cancel</b>	Close this dialog box without saving changes.
<b>Apply</b>	Apply changes without closing the dialog box. Changes made with the Apply button cannot be canceled by clicking the Cancel button.

## Editor Preferences Tab

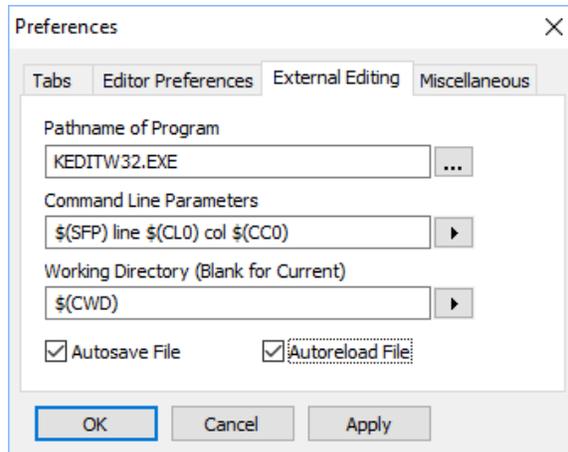
The Editor Preferences tab, illustrated below, allows you to configure the environment to be used by an external editor.



External Editing Tab	
Use this option	To
<b>Maximum Record Length</b>	<p>Like the <b>SOURCE-RECORD-MAX-LENGTH</b> configuration option, specifies the maximum source record length. The value is a number from 80 to 65000. The default is 1024. The maximum source record length is established at the beginning of a compilation and cannot be changed during that compilation (see “Source Format” in Chapter 1: Language Structure of the RM/COBOL Language Reference Manual). Source records can vary in length from 0 characters up to the maximum source record length specified. Longer records are truncated to the maximum source record length; when such truncation occurs, the compiler reports at the end of compilation how many records were truncated.</p> <ul style="list-style-type: none"> <li>• If a value less than 80 is given for this keyword, the maximum source record length is set to 80 with no warning or error diagnostic.</li> <li>• If a value greater than 65000 is given for this keyword, a configuration value error (E0009) occurs.</li> </ul>
<b>Right Margin after Column</b>	<p>Like the <b>INITIAL-MARGIN-R</b> configuration option, this value specifies the position of the default right margin. This position determines the end of the program-text area and the beginning of the Identification area, if any. See the explanation of fixed-form reference format in the <i>RM/COBOL Language Reference Manual</i> for more information on margin R. The value for this keyword is a number specifying the column after which margin R is located. The minimum value is 72 and the maximum value is the same as the maximum source record length.</p> <p>Margin R can also be changed at any time during the compilation with the RM/COBOL implementer-defined directive <b>IMP MARGIN-R</b>.</p>
<b>Use Default Right Margin and Record Length</b>	<p>Uses the default values for the above fields. Uncheck this box to customize the maximum record length and the right margin.</p>

## External Editing Tab

The External Editing tab, illustrated below, allows you to configure the environment to be used by an external editor.



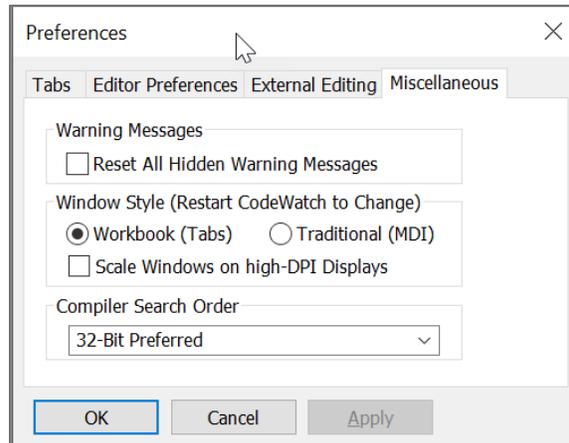
External Editing Tab	
Use this option	To
<b>Pathname of Program</b>	<p>Specify the full pathname of the external editor program to invoke. This should be the complete pathname of the executable file, but can also be the name of a DOS .bat file, if you are invoking a DOS-based external editor. Click the ellipsis button to display a Select Editor Program File dialog box in order to search for a program.</p> <p><b>Note</b> An external editor has its own character set conventions. Care should be taken in choosing the external editor so that its character set matches the workspace character set.</p>
<b>Command Line Parameters</b>	<p>Specify the command line to be passed to the external editor. The contents of this field will depend on your chosen editor. Various substitution symbols are available to pass the current context (file, cursor line, and position) to the external editor. Press the arrow button to display a submenu of substitution symbols. Available symbols include the following:</p> <ul style="list-style-type: none"> <li>• \$(SFP) The pathname of the current source file.</li> <li>• \$(SFN) The current source filename and extension.</li> <li>• \$(SFD) The directory that contains the current source file.</li> <li>• \$(SFE) The extension of the current source file, including the period.</li> <li>• \$(CL0) The line number containing the cursor, where 0 is the first line.</li> <li>• \$(CC0) The column number containing the cursor, where 0 is the left-most column.</li> <li>• \$(CL1) The line number containing the cursor, where 1 is the first line.</li> <li>• \$(CC1) The column number containing the cursor, where 1 is the left-most column.</li> <li>• \$(CWD) The current working directory.</li> <li>• \$\$ A single "\$" symbol.</li> </ul>

External Editing Tab	
Use this option	To
<b>Working Directory</b>	<p>Specify the working directory that is set for the external editor. Leave this field blank to use the directory where CodeWatch is running. Press the arrow button to display a submenu of substitution symbols. Available symbols include the following:</p> <ul style="list-style-type: none"> <li>• \$(CWD) The current working directory. This is the default.</li> <li>• \$(EWD) The current “execution” directory, as specified in the <a href="#">Inspect/Modify dialog box</a> (see page 66) when you choose Workspace   Properties command from the File menu.</li> <li>• \$\$ A single “\$” symbol.</li> </ul>
<b>Autosave File</b>	<p>Determine how CodeWatch behaves when you invoke an external editor on a file that you have changed, but not saved. If this option is selected, the file is automatically saved. When this option is not selected, the file is not automatically saved, and CodeWatch will present the following prompt in a message box:</p> <pre>This file has been edited but has not been saved. Save it now?</pre> <p>Choose Yes to save the file and start the external editor. Choose No to start the external editor without saving your changes. Choose Cancel to return to CodeWatch without invoking the external editor.</p>
<b>Autoreload File</b>	<p>Determine how CodeWatch behaves when you return to CodeWatch from another application. This applies whether you are returning from an external editor or from any other program. If this option is selected, CodeWatch will automatically reload any file open in the external editor that has been changed outside CodeWatch, but only if there are no unsaved changes.</p> <p>If this option is not selected, and you return to CodeWatch after making changes to a file outside CodeWatch, you will see the following message box:</p> <pre>The file "filename.cbl" was modified outside CodeWatch. Would you like to reload this file?</pre> <p>Choose Yes to reload the file, and choose No to keep working without reloading the file.</p> <p>If this option is selected, and you return to CodeWatch after making changes to a file outside CodeWatch and you have not changed the file within CodeWatch, the changes will automatically be loaded into the CodeWatch Source Code Editor. If, however, you have made changes in CodeWatch, you must choose which set of changes to keep. The following message box will appear:</p> <pre>The file "filename.cbl" was modified outside CodeWatch, and there are also unsaved changes in the CodeWatch editor. Would you like to reload this file and lose any changes you made since the last time this file was saved?</pre> <p>Choose Yes to reload the file and discard your changes. Choose No to continue working in CodeWatch.</p> <p><b>Note</b> In either of the above cases, you can choose No, and then use the Save Copy As command from the File menu to save a copy of the version of the file in the CodeWatch Source Code Editor. You can then close and reopen the file to load the new version.</p>
<b>OK</b>	Apply any changes made above and close this dialog box.
<b>Cancel</b>	Close this dialog box without saving changes.

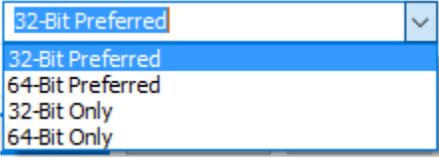
External Editing Tab	
Use this option	To
<b>Apply</b>	Apply changes without closing the dialog box. Changes made with the Apply button cannot be canceled by clicking the Cancel button.

## Miscellaneous Tab

The Miscellaneous tab, illustrated below, allows you to configure several options:



External Editing Tab	
Use this option	To
<b>Reset All Hidden Warning Messages</b>	Some CodeWatch warnings have an option to suppress future messages. To see these again, check the box and press the <b>OK</b> or <b>Apply</b> button.
<b>Window Style</b>	<p>Determines how the document window functions. Select the Workbook Style to use tabs across the bottom of the window, which is preferred but takes up some space in the window. Select Traditional to use the standard MDI style. In either style, a list of open document windows is added to the bottom of the Window menu, and this (or Alt+Tab) can be used to move between Windows.</p> <p>Windows can be cascaded, tiled, or split when either window style is in effect.</p> <p>CodeWatch must be restarted before changes to this option take effect.</p>
<b>Scale Windows on high-DPI Displays</b>	<p>Determines if CodeWatch will be displayed at its normal resolution, or automatically scaled in accordance with the scale factor set in the Microsoft Windows system Display settings.</p> <p>See <a href="#">High-DPI Display Considerations</a> for a discussion of this setting.</p> <p>This option is only available on Microsoft Windows 8.1 and later and only has an effect if the main display has a scale factor other than 100%.</p> <p>CodeWatch must be restarted before changes to this option take effect.</p>

External Editing Tab	
Use this option	To
<b>Compiler Search Order</b>	<p>Determines whether the 32-bit or 64-bit compiler will be used to build the workspace. Available options are:</p>  <p>“32-Bit Preferred” is the default – CodeWatch will use the 32-bit compiler if installed, or the 64-bit compiler if the 32-bit compiler is not present.</p> <p>“64-bit Preferred” will use the 64-bit compiler by default, or the 32-bit compiler if the 64-bit compiler is not present.</p> <p>The final two options can be used to force a particular compiler to be used. An error occurs if the specified compiler (or no compiler) is not available.</p> <p>This setting can also be changed by editing the .INI file: see <i>Choosing the 32-bit or 64-bit RM/COBOL Compiler</i>.</p>

---

## Basic Editing Techniques

This section provides an overview of the most common and basic editing tasks in CodeWatch, including the following:

- [Moving around in a file](#) (see the next topic)
- [Selecting text](#) (see page 118)
- [Inserting new text](#) (see page 120)
- [Overwriting or replacing text](#) (see page 121)
- [Deleting text](#) (see page 121)
- [Correcting and editing text](#) (see page 121)
- [Cutting, copying, and pasting text](#) (see page 123)
- [Using the Windows Clipboard to duplicate or move text](#) (see page 123)

You may also find it helpful to refer to the mouse and keyboard techniques, as described in [How to Move with the Mouse](#) (on page 116) and [How to Move with the Keyboard](#) (on page 117).

### Moving Around in a File

To move around and control the part of the file that is visible in a Source Code/Edit window, you can use the mouse or the keyboard. Both techniques are discussed in this section, starting with the mouse-based choices. First, however, let’s review the insertion point (also known

as the cursor or caret). You use the insertion point to specify the location you want to edit in the file.

## **Insertion Point**

The insertion point indicates your current position in the file. The insertion point on the status bar in CodeWatch is referred to as the “Edit point.”

The visual appearance of the insertion point corresponds to the type of text-entry mode: Insert mode, Overtyping mode, and Selection mode. You are in Selection mode when you select one or more characters by dragging the mouse or using the keyboard. In all other cases, you are in Insert or Overtyping mode. The **Ins** (Insert) key toggles between Insert mode and Overtyping mode. An indicator on the status bar is blank when you are in Insert mode, and indicates **OVR** when you are in Overtyping mode.

For more details on the appearance of the insertion point and these three text-entry mode options, see [Selecting Text](#) (on page 118).

You can move the insertion point anywhere within the text of a line on the screen by positioning the I-beam mouse pointer to the desired location and clicking the left mouse button. You can also move the insertion point using the direction keys on the keyboard, as described in [Table 3](#) on page 117.

**Note** There is one essential difference between using the direction keys on the keyboard to move or page through a file and using the mouse to move or scroll through a file. When you use the keyboard, the insertion point will travel with you as you move through the file. When you use the mouse, however, the insertion point does not change location and remains where it was before you moved. For example, if you use the mouse to view another part of your file and then start entering text, CodeWatch returns you to your starting place and anything you have typed is entered at the original insertion point location.

## **How to Move with the Mouse**

The primary mouse-based navigation tools are the horizontal and vertical scroll bars, and the mouse wheel (if your mouse is so equipped). The boxes within the scroll bars are called scroll boxes, and the arrows at either end of each scroll bar are called scroll arrows.

**Note** To move the insertion point to a position in the file that is not currently visible, use the scroll bars to make the position visible, then position the I-beam mouse pointer and click the mouse button.

## **Using the Vertical Scroll Bar**

To move through your file one line at a time, click on the scroll arrow at the top or bottom of the vertical scroll bar. Click the bottom arrow to scroll down—or forward—through a file and the top arrow to scroll up—or backward. To move through a file one full screen at a time, click anywhere within the scroll bar—above the scroll box to scroll up, or below the box to scroll down. You can also quickly and continuously scroll forward and backward through a file by clicking and holding down the mouse button over the scroll arrow.

To move long distances through a file, you can drag the scroll box to the appropriate position in the scroll bar. The scroll box position on the vertical scroll bar represents your relative position in the document.

## Using the Horizontal Scroll Bar

If you are working with a file that is too wide to fit on the screen, you can use the scroll arrows at either end of the horizontal scroll bar to move right and left. Horizontal scrolling is much like vertical scrolling. Use the scroll arrows to move short distances, click on either side of the scroll box to move by a full screen, or drag the scroll box to reposition the window manually.

## How to Move with the Keyboard

With the keyboard commands, you can move short distances with the arrow keys, move by words or paragraphs, move by one screen at a time, and move to the beginning or end of a file. [Table 3](#) summarizes the keystrokes used to move the insertion point in a file.

**Note** Pressing an arrow key moves the insertion point one character or one line in the direction of the arrow. If the insertion point is already at that edge of the window, the window scrolls to keep the insertion point visible.

**Table 3: Direction Keys**

Direction Keys to Move the Insertion Point or Cursor	
Action	Press
Left one character.	Left Arrow
Right one character.	Right Arrow
Left one word.	Ctrl+Left Arrow
Right one word.	Ctrl+Right Arrow
To the first non-blank character of the current line, or the beginning of the current line.	Home
To the end of the current line (to the first trailing blank).	End
Up one line.	Up Arrow
Down one line.	Down Arrow
To the first line of the screen.	Ctrl+Page Up
To the last line of the screen.	Ctrl+Page Down
Up one screen.	Page Up
Down one screen.	Page Down
To the beginning of the file.	Ctrl+Home
To the end of the file.	Ctrl+End
To one tab stop to the right.	Tab

Direction Keys to Move the Insertion Point or Cursor	
Action	Press
To one tab stop to the left.	Shift+Tab (or Back Tab)
To the next item matching the last search item.	F3
To the previous item matching the list search item.	Shift+F3
To the next word that matches the word under the cursor.	Ctrl+F3
To the previous word that matches the word under the cursor.	Ctrl+Shift+F3
Into the next pane if this window is split.	Ctrl+F6
Into the previous pane if this window is split.	Ctrl+Shift+F6

Direction Keys To Move the Window over the Document	
Action	Press
Scroll the window up one line.	Ctrl+Down Arrow
Scroll the window down one line.	Ctrl+Up Arrow

## Selecting Text

Before you can edit a given portion of text, you first have to select it. When you select text, the CodeWatch Source Code Editor enters Selection mode, and the insertion point is displayed highlighting the selected text in reverse video (white text and graphics on a black background rather than the normal black on white).

**Note** Throughout this user's guide, this reverse video effect is referred to as selected text or highlighted text.

You can select one or more characters of text in order to edit the entire selection as a single unit. To cancel a selection (and return to Insert or Overtyping mode), position the I-beam mouse pointer outside the selected text and click the mouse button, or use any of the direction keys described previously in [Table 3](#). See also [Inserting New Text](#) (on page 120) and [Overwriting or Replacing Text](#) (on page 121).

You can select text in many ways using the mouse or the keyboard. Each approach offers its own advantages.

### **How to Select Text with the Mouse**

The following sections describe ways to select text using the mouse. To cancel a selection using the mouse, click anywhere outside the selection in the window, or press the **Esc** key.

## Selecting a Word

To select an entire word, hold down the **Ctrl** key, and single-click in the word. The entire word will be highlighted. Pressing **Shift+Ctrl+W** has the same effect.

Double-clicking also selects a word. In addition, if the program is running, the [Inspect/Modify dialog box](#) (see page 66) is displayed and the word is displayed as if it were an identifier.

## Selecting Entire Lines

To select a single line, click in the gutter to the left of the line of text. The entire line will be highlighted. Dragging the mouse in the gutter selects multiple lines.

To select multiple lines, position the I-beam mouse pointer and drag the insertion point so that text from at least two lines is selected.

## Selecting the Entire Document

To select the entire document, hold down the **Ctrl** key and click in the gutter, choose **Select All** from the **Edit** menu, or press **Ctrl+A** anywhere in the file.

## Selecting Multiple Characters or Lines

To select one or more characters, position the I-beam mouse pointer and drag over the desired text. Begin by positioning the I-beam mouse pointer just to the left of the text you wish to select. Press and hold down the left mouse button, and without releasing the button, drag to the right until you have highlighted the desired text. Release the mouse button. The characters will remain highlighted. You can use dragging to select any size block of text, from a single character to an entire file.

Similarly, dragging in the gutter will select more than one entire line.

## Block Selection

To select a rectangular block of text, hold down the **Alt** key, and click in one corner of the selection. While holding down the mouse key, drag the mouse to the opposite corner (it need not be in the window—the Source Code/Edit window will scroll if necessary). All text in the rectangle will be selected.

## Extending a Selection

To extend a selection, hold down the **Shift** key while dragging with the mouse.

- To select all text between the cursor and another location on the screen, hold down the **Shift** key and click with the mouse. All text between the cursor position and the point where the mouse was clicked will be selected.
- If text is selected, you can reshape the selection by holding down the **Shift** key and clicking with the mouse. All text between the origin of the selection and the location where you clicked will be selected.

## How to Select Text with the Keyboard

CodeWatch provides techniques for selecting text with the keyboard, as listed in [Table 4](#). You can hold down combinations of the **Shift** and **Ctrl** keys to turn the direction keys (described in [Table 3](#) on page 117) into selection keys. Place the insertion point where you want to begin your selection, and then press and hold down **Shift** and/or **Ctrl** while using any of the keys listed. You will simultaneously move the insertion point and highlight the text.

**Table 4: Text Selection Keys**

Text Selection Keys	
To extend the selection	Press
One character at a time to the left or right.	Shift+Left Arrow or Shift+Right Arrow
One line of text up or down.	Shift+Up Arrow or Shift+Down Arrow
Left one word.	Ctrl+Shift+Left Arrow
Right one word.	Ctrl+Shift+Right Arrow
To the first nonblank character on the current line.	Shift+Home
To the beginning of the current line.	Shift+Home again
To the end of the current line (to the first trailing blank).	Shift+End
Text to the beginning of the file.	Ctrl+Shift+Home
Text to the end of the file.	Ctrl+Shift+End
Up one screen.	Shift+Page Up
Down one screen.	Shift+Page Down
To the entire file	Ctrl+A

## Inserting New Text

Inserting text is as easy as typing, but you first must be in Insert mode. In Insert mode, the insertion point appears on the screen as a blinking vertical line. The **Ins** (Insert) key is used to switch between Insert mode and Overtyping mode (described in the following topic, *Overwriting Text*).

As you enter new characters in Insert mode, they are inserted to the left of the insertion point and the insertion point and all text following the insertion point move to the right. If necessary, the Source Code/Edit window scrolls to keep the insertion point visible.

## Overwriting or Replacing Text

In Overtyping mode, the insertion point appears as a blinking reverse video block cursor. As you type, CodeWatch overwrites the character highlighted by the cursor. If necessary, the Source Code/Edit window scrolls to keep the insertion point visible.

### Overwriting Selected Text

You can overwrite any selected text whether you are in Insert or Overtyping mode. When you select one or more characters with the mouse or keyboard, you enter Selection mode. In Selection mode, the selected text appears on the screen in reverse video. The first character you type replaces all the selected text immediately, returns to the mode that was active before you selected the text, and moves the insertion point one character to the right. If necessary, the Source Code/Edit window scrolls to keep the insertion point visible.

## Deleting Text

The easiest way to delete a character is to place the insertion point to the right of the character and press **Backspace**. CodeWatch will delete one character to the left of the insertion point, pulling characters at the right of the insertion point one space to the left. If you prefer, you can place the insertion point to the left of the character you want to delete and press **Del**. In this case, CodeWatch will delete the character to the immediate right and move all following characters to the left.

To delete several characters at once, you can select (or highlight) them and then press either **Backspace** or **Del** (Delete). CodeWatch will delete the selection and fill in the gap by pulling the remaining text in the Source Code/Edit window up and to the left. You can also delete text with the **Cut** command from the **Edit** menu. See [Cutting, Copying, and Pasting Text](#) (on page 123).

Keyboard shortcuts are also available for deleting text. For example, instead of pressing **Backspace** to delete characters one at a time, you can press **Ctrl+Del** to delete the word containing the cursor. [Table 5](#) describes these editing shortcut keys.

## Correcting and Editing Text

As described in the previous sections, by entering the appropriate mode, you can insert new text, replace old text, or replace selected text. You can insert the typed character(s) at the insertion point (in Insert mode), replace the character(s) at the insertion point (in Overtyping mode), or replace the current selection (in Selection mode).

The CodeWatch Editor also provides several editing keys to aid in correcting text, as shown in [Table 5](#).

**Table 5: Editing Keys**

Editing Keys		
To	In This Mode	Press
Change the selected text to uppercase.	Selection	Ctrl+U
Change the selected text to lowercase.	Selection	Ctrl+Shift+U

Editing Keys		
To	In This Mode	Press
Delete the selected text. Delete the character preceding the insertion point.  <b>Note</b> If the insertion point is at the beginning of a line, this action joins the current line to the end of the previous line.	Selection Insert, Overtyping	Backspace
Delete the selected text. Delete the character following the insertion point. Delete the highlighted character.	Selection Insert Overtyping	Del (Delete)
Replace the selected lines with a blank line if one or more lines are selected. If only part of a single line is selected, delete the selected characters and split the line at the insertion point. Delete the selected text and move to the first non-blank character on the next line. Split the line at the insertion point. Move to the first non-blank character on the next line.	Selection, Insert  Selection  Insert Overtyping	Enter
Delete all text on the current line.	All	Ctrl+Backspace
Delete all text from insertion point to the end of the line.	All	Ctrl+Del
Delete the current line.	All	Ctrl+L
If the insertion point is within a word, delete from the beginning of the word to the beginning of the next word. If the insertion point is not within a word, delete from the insertion point to the beginning of the next word.	All	Ctrl+Del
Shift the text from the insertion point to the end of the line right to the next tab stop by inserting blanks in the vacated positions.  <b>Note</b> If the selection encompasses more than one line, all lines in the selection are shifted. This is a convenient way of indenting a block of code.	All	Tab
Shift the text from the insertion point to the end of the line left to the previous tab stop, provided there are no non-blank characters between the insertion point and the previous tab stop.  <b>Note</b> If the selection encompasses more than one line, all lines in the selection are shifted. This is a convenient way of unindenting a block of code.	All	Shift+Tab
Split the current line at the insertion point.  <b>Note</b> If the insertion point is at the beginning of a line, this action inserts a blank line before the current line. If the insertion point is after the last non-blank character, this action inserts a blank line after the current line.	Insert	Enter

## Cutting, Copying, and Pasting Text

You can also delete text with the **Edit | Cut** command. Unlike the previously discussed deletion techniques, this method stores the deleted text in the Windows Clipboard, which is a temporary holding area for data that is shared across all Windows programs. You can then paste the Clipboard's contents anywhere in your file, into another file, or even into another program.

In general, you copy or move text by selecting the text and then choosing the location where you want to insert, or paste, it. For more information, see [Using the Windows Clipboard to Duplicate or Move Text](#) (on page 123).

Instead of typing a given block of text when you need a copy, you can use the **Edit | Copy** command to copy the text to the Clipboard, followed by the **Edit | Paste** command to paste the text in one or more locations in your file. A copy of the text will be inserted just to the left of the insertion point, leaving the insertion point at the end of the pasted block. With the **Paste** command, if you are in Selection mode, the pasted text replaces the selected text; if you are not in Selection mode, the text is inserted. The contents of the Clipboard remain unchanged.

You can also use keyboard shortcuts for cutting, copying, and pasting text. The keys or key combinations appear to the right of the appropriate commands. [Table 6](#) summarizes the keyboard shortcuts for the Edit menu.

**Table 6: Edit Menu Shortcut Keys**

Edit Menu Shortcut Keys	
To select the	Press
Undo command	Ctrl+Z or Alt+Backspace
Redo command	Ctrl+Y or Shift+Alt+Backspace
Cut command	Ctrl+X or Shift+Del
Copy command	Ctrl+C or Ctrl+Ins
Paste command	Ctrl+V or Shift+Ins
Clear command	Del
Delete the line containing the cursor	Ctrl+ L
Cut the line containing the cursor	Ctrl+Shift+L

## Using the Windows Clipboard to Duplicate or Move Text

The Windows Clipboard is a temporary holding area used to transfer text either within a file, between files, or between applications. You use the Cut, Copy, and Paste commands from the Edit menu to move or duplicate information.

First, select text in the source file and then do one of the following:

- Choose **Edit | Cut** or press **Ctrl+X** or **Shift+Del** to move the selected text to the Clipboard. This action removes the selection from the file.

- Choose **Edit | Copy** or press **Ctrl+C** or **Ctrl+Ins** to copy the selected text to the Clipboard. This action leaves the original text in the file.

Next, position the insertion point in the destination (or target) file and do the following:

- Choose **Edit | Paste** or press **Ctrl+V** or **Shift+Ins** to insert the selected text. You can do this any number of times.

Only one Clipboard is available to all Windows programs, so any cut or copy operation in any program will replace the contents of the Clipboard.

# Chapter 5: Building Programs with CodeWatch

This chapter provides information on how CodeWatch compiles source files.

## Compiling

There are several ways to compile a file under CodeWatch.

Compiling		
To Compile	Choose	Or click
The file that is currently being edited. Note that any changes must be saved to disk before the file is being compiled.	Compile File on the Build menu or press Ctrl+F7.	 Compile File
All files that have changed since the last compilation or that have out-of-date object files. The source files that will be compiled are indicated by a checkmark on the Files tab in the Workspace window. For further information, see <a href="#">Files tab</a> (on page 49).	Compile Changes on the Build menu or press F7.	 Compile Changes
All source files in the workspace.	Compile All on the Build menu or press Ctrl+Shift+F7.	 Compile All

### Compiling Only Changed Files

The **Compile Changes** command on the [Build menu](#) (see page 76) recompiles source files that are determined to require recompilation. Source files are deemed to require recompilation if any of the following conditions are true:

- The source file has never been compiled, that is, there is no object file for the source file.
- The source file has been edited since the last compilation. This is determined by comparing the date of the source file to that of the object file. If the source file is newer than the object file, recompilation is required.

- A copy file used by the source file has been edited since the last compilation. This is determined by comparing the date of the copy file to that of the object file. If the copy file is newer than the object file, recompilation is required.
- An RM/COBOL compilation option has been changed in the workspace that affects the generated program. This includes any of these Compile Command Options: /D, /G, /H, /N, /O, /Q, /S, /U, /V, /Y, /Z, /2 and /7.

## Setting Compilation Options

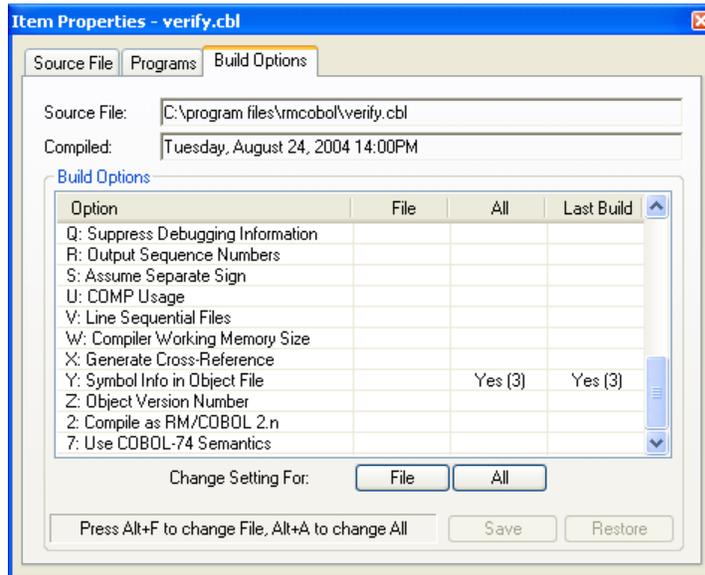
The Build Options property tab of the [Item Properties dialog box](#) (see page 91), illustrated below, allows you to view and change the compilation options for a particular file or the workspace.

To display the Build Options property tab, make sure the cursor is in the Source Code/Edit window and do one of the following:



- From the **Build** menu, click **Options**.
- Click the **Options** toolbar button.
- Press **Alt+F7**.
- From the **File** menu, choose **Item Properties**, and then click the **Build Options** tab.

For a description of the options on this tab, see [Build Options property tab](#) (on page 93).



Note that CodeWatch will recommend that a file be recompiled under the following circumstances:

- The setting in the **File** column is not blank and does not match the setting in the **Last Build** column.
- The setting in the **File** column is blank (that is, not specified), the setting in the **All** column is not blank, and that setting does not match the setting in the **Last Build** column.

In other words, the **File** setting has the highest priority, and the **All** setting has the second highest priority.

[Table 7](#) lists the RM/COBOL Compile Command Options that can be set in the **Build Options** property tab of the Item Properties dialog box. (For more information on the RM/COBOL compilation options, see Chapter 6: *Compiling of the RM/COBOL User's Guide*.)

Note the following:

- A check (✓) in the Dep? (Dependent) column indicates that a change to this option will mark the source file for recompilation.
- The Default Value column indicates the value that will be used if this option is not specified.
- Italics (for example, *Numeric Value*) indicate that you will be prompted to enter this value.
- If you select conflicting options (for example, both B=Yes and V=Yes or 2=Yes and 7=No), CodeWatch will issue a warning and require that the error be corrected.

**Table 7: RM/COBOL Compile Command Options Available**

RM/COBOL Compile Command Options			
Option Description	Dep?	Default Value	Other Values
A: Produce Allocation Map		No	Yes
B: Binary Sequential Files	✓	No	Yes
C: Suppress Copy File Display		No	Yes
D: Compile WITH DEBUG MODE	✓	No	Yes
E: Errors Only Disk Listing		No	Yes
F: Flag COM1		No	Yes
F: Flag COM2		No	Yes
F: Flag EXTENSION		No	Yes
F: Flag HIGH		No	Yes
F: Flag INTERMEDIATE		No	Yes
F: Flag OBSOLETE		No	Yes
F: Flag SEG1		No	Yes
F: Flag SEG2		No	Yes
G: Configuration Override File	✓	None	<i>Filename</i>
H: Configuration Supplement File	✓	None	<i>Filename</i>

RM/COBOL Compile Command Options					
Option Description	Dep?	Default Value	Other Values		
L: Produce Disk Listing		No	Yes	<i>Directory</i>	
M: Automatic Input Conversion	✓	No	Yes		
N: Suppress Object Generation	✓	No	Yes		
O: Object Directory	✓	Default	<i>Directory</i>		
P: Print Compilation Listing		No	Yes		
Q: Suppress Debugging Information	✓	No	Yes		
R: Output Sequence Numbers		No	Yes		
S: Assume Separate Sign	✓	No	Yes		
U: COMP Usage	✓	Default	Binary	Packed	Display
V: Line Sequential Files	✓	No	Yes		
W: Compiler Working Memory Size		Default	<i>Numeric Value</i>		
X: Generate Cross-Reference		No	Yes		
Y: Symbol Info in Object File	✓	No	Yes(3)	Yes(2)	Yes(1)
Z: Object Version Number	✓	Latest	“Version <i>n</i> ” where <i>n</i> specifies object version level number 9 to 15.		
2: Compile as RM/COBOL 2. <i>n</i>	✓	No	Yes		
7: Use COBOL-74 Semantics	✓	No	Yes		

**Note** For more information on the RM/COBOL compilation options, see Chapter 6: *Compiling of the RM/COBOL User’s Guide*.

## The RMPATH and Environment Variables

The compiler uses the contents of the RMPATH environment variable to locate copy files. The content of this path variable is saved in the workspace, and can be set from the Environment tab of the Workspace Properties dialog box.

To set the RMPATH environment variable:

1. From the File menu, choose Workspace | Properties.
2. Click the **Environment** tab, and then click the **Source** tab.
3. Add the names of the directories that contain your copy files to the list. To add an item to the list, click in the list box and begin typing.

To rearrange the order of the directories, you can drag entries with the mouse, or use the Move Up and Move Down buttons.

Compilation under CodeWatch is also affected by the contents of the Environment tab. Synonyms may be assigned using this tab, and any of the environment variables described in Chapter 2: *Installation and System Considerations for UNIX* and Chapter 3: *Installation and System Considerations for Windows* of the *RM/COBOL User's Guide*, may be set in this way.

## Using 32-Bit and 64-Bit Compilers

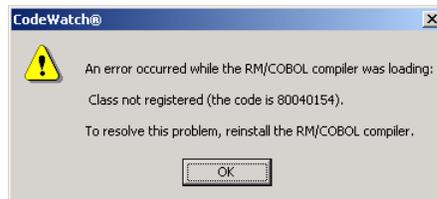
In addition to the 32-bit Windows development system, RM/COBOL now offers a new 64-bit Windows development system, including a 64-bit compiler.

CodeWatch will automatically locate an installed compiler, regardless of whether a 32-bit or 64-bit development system is installed. If both are installed, CodeWatch will prefer the 32-bit compiler for backward-compatibility.

To force CodeWatch to choose a specific compiler, or to alter the search sequence, see [Compiler Search Order](#).

## Locating the Compiler

The RM/COBOL compiler is located by Windows and does not need to be in the system search path. If, for some reason, CodeWatch cannot find the compiler, you will see a message error box like the following:



To resolve this problem, reinstall the RM/COBOL compiler. You can also try to reregister the server:

- Open a command window.
- Change into the directory where you installed the compiler. By default, on 64-bit Windows systems, this is:

```
32-bit - c:\Program Files (x86)\Micro Focus\RM\RMCOBOLv12
64-bit - c:\Program Files\Micro Focus\RM\RMCOBOLv12-64
```

- On 32-bit Windows system, this is:

```
c:\Program Files\Micro Focus\RM\RMCOBOLv12
```

- Issue this command: **rmcobol /regserver**

Then, retry the compilation. There is no need to restart CodeWatch.

For more information about registering the compiler and runtime executables, see Chapter 3: *Installation and System Considerations for Microsoft Windows* of the *RM/COBOL User's Guide*.

## Compiler Output

When one of the **Build** commands is given, CodeWatch takes the following actions:

- The Log window is cleared.
- The Build Results tab in the Data window is cleared.
- You are asked to save any files that are about to be compiled and are being edited, and that have been changed since they were last saved or opened.
- Each file in the list of files that are awaiting compilation is compiled, using the options that are currently set for that file and for the workspace.
  - The output of the compiler is sent to the Log window.
  - As each file is compiled, the results are also logged into the Build Results tab of the Data window.
- If a file is open in the editor, the file is reloaded and merged with listing information from the program file.
- Files with errors or warnings are highlighted in the Build Results tab of the Data window.

# Appendix A: Additional Configuration Options

CodeWatch includes several configuration options and tuning parameters that are not set from within the program. Instead, these options are read from a file called **rmcw.ini**. This file is created in the current user's application data directory when CodeWatch is run for the first time. The **rmcw.ini** file can be edited with any text editor, including the CodeWatch Source Code Editor. To edit this file in CodeWatch, choose Open from the File menu and enter **%APPDATA%\Micro Focus\RM\CodeWatch\rmcw.ini**.

**WARNING** The **rmcw.ini** file contains other keys that are not described below. These keys are for internal use only, and their values should be altered only under the guidance of Micro Focus Customer Care.

---

## Runtime Options

The configuration settings for several RM/COBOL Runtime Command Options are discussed in the following topics.

### Setting the RM/COBOL Runtime Registry Key

The RM/COBOL runtime system uses the following registry key to read configuration information out of the Windows registry. The `RuntimeRegistryKey` option needs to be set only if you have renamed your **runcobol.exe** program. In that case, the specified key should be the same as the name of the runtime, without a directory or the **.exe** extension.

```
[Defaults]
RuntimeRegistryKey=key
```

This value is read whenever the RM/COBOL runtime is started from within CodeWatch. If not specified, **runcobol** is used as the key.

### Setting the RM/COBOL Runtime Configuration .exe File

When CodeWatch is running, a configuration file is normally specified by using the `/C` and `/X` Runtime Command Options. However, an RM/COBOL configuration file also can be

attached to **runcobol.exe** by using the Attach Configuration utility (**rmattach**) program, as described in Appendix G: *Utilities* of the *RM/COBOL User's Guide*. When **runcobol.exe** is started outside CodeWatch, the attached configuration file is used to modify the behavior of the RM/COBOL runtime.

CodeWatch, however, runs programs without using **runcobol.exe**. It is possible to specify the location of an **runcobol.exe** file that has an attached configuration file to emulate the behavior of the standalone runtime.

```
[Defaults]
RuntimeConfigurationEXE=program-file
```

If such a configuration **.exe** file is specified, the name of the file is also used as the default registry key unless the `RuntimeRegistryKey` option is specified. (For more information, see [Setting the RM/COBOL Runtime Registry Key](#) on page 131.) The directory containing the **.exe** file is used to locate the runtime license vault unless the `RuntimeLicensePathname` option is specified, as described in the following topic, [Selecting a Runtime License Vault](#).

## Selecting a Runtime License Vault

Because you may have more than one licensed runtime system installed, there may be times when you need to specify that CodeWatch use a particular license vault.

```
[Defaults]
RuntimeLicensePathname=directory
```

The `RuntimeLicensePathname` option allows you to specify a directory that contains a **license.vlt** file. That license vault will be used for all subsequent debugging sessions.

This option is not required if you are using the development system for debugging. It is required only if you have multiple runtime systems installed, or if you have made multiple copies of the runtime.

---

## Compiler Options

The configuration settings for several RM/COBOL Compile Command Options are discussed in the following topics.

### Setting the RM/COBOL Compiler Registry Key

The RM/COBOL compiler uses the following registry key to read configuration information out of the Windows registry. The `CompilerRegistryKey` option needs to be set only if you have renamed your **rmcobol.exe** program. In that case, the specified key should be the same as the name of the compiler, without a directory or the **.exe** extension.

```
[Defaults]
CompilerRegistryKey=key
```

This value is read whenever the RM/COBOL compiler is started from within CodeWatch. If not specified, **rmcobol** is used as the key.

## Setting the RM/COBOL Compiler Configuration .exe File

When compiling, a configuration file is normally specified by using the /G or /H Compile Command Options. However, an RM/COBOL configuration file also can be attached to **rmcobol.exe** by using the Attach Configuration utility (**rmattach**) program, as described in Appendix G: *Utilities* of the *RM/COBOL User's Guide*. When **rmcobol.exe** is started outside CodeWatch, the attached configuration file is used to modify the behavior of the RM/COBOL compiler.

CodeWatch, however, compiles without using **rmcobol.exe**. It is possible to specify the location of an **rmcobol.exe** file that has an attached configuration file to emulate the behavior of the standalone compiler.

```
[Defaults]
CompilerConfigurationEXE=program-file
```

If such a configuration .exe file is specified, the name of the file is also used as the default registry key unless the CompilerRegistryKey is specified. See the previous topic, [Setting the RM/COBOL Compiler Registry Key](#).

## Choosing the 32-bit or 64-bit RM/COBOL Compiler

In addition to the 32-bit Windows development system, RM/COBOL offers a 64-bit Windows development system, including a 64-bit compiler.

CodeWatch will automatically locate an installed compiler, regardless of whether a 32-bit or 64-bit development system is installed. If both are installed, CodeWatch will prefer the 32-bit compiler for backward-compatibility.

It is possible to force CodeWatch to prefer the 64-bit compiler by using the “Compiler Search Order” option on the Miscellaneous tab in the Preferences dialog box. See the [Compiler Search Order](#) dialog box topic for a description.

The option can also be changed by making a manual entry in the CodeWatch configuration file located here:

```
[Defaults]
CompilerSearchOrder=0
```

The **CompilerSearchOrder** values may be set to:

- 0 Look for the 32-bit compiler, then the 64-bit compiler
- 1 Look for the 64-bit compiler, then the 32-bit compiler
- 2 Look for only the 32-bit compiler, ignore any 64-bit compiler
- 3 Look for only the 64-bit compiler, ignore any 32-bit compiler

The new setting takes effect once CodeWatch is restarted.

## Miscellaneous Options

The configuration settings for several miscellaneous options are discussed in the following topics.

### Changing the Log Window Font

The default font used in the Log window is the same font that is used in the Source Code Editor window

```
[Defaults]  
LogFont=font [, size]
```

The following example illustrates how you can use this option to change the default font to Lucida Console, 9 point, which would allow more lines to be displayed in the Log window:

```
LogFont=Lucida Console,9
```

In order to see this change, however, you must close and reopen the Log window.

### Setting the Log Window Size

Use this option to change number of lines retained in the Log window.

```
[Defaults]  
MaxLogLines=value
```

By default, only the last 150 lines are retained.

### Controlling Dependency Checking

When you switch from CodeWatch to another window, and then switch back, CodeWatch checks for source and copy files that may have been changed. To prevent a delay from being noticeable, by default, CodeWatch checks ten files at a time. If you have a very large workspace and notice a significant delay when switching from another window (for example, an external editor) back to CodeWatch, you may wish to reduce this number.

```
[Defaults]  
IdleCheckDependencyCount=value
```

Reducing this value may make CodeWatch more responsive when switching windows, but it should not have an effect during normal operations.

# Appendix B: Summary of Enhancements

This appendix provides a summary of the new features and changes in the various releases of CodeWatch. The enhancements and changes for the most recent release described by this document are listed first.

**Note** The information in this appendix is historical. It was accurate at the time written for the specific version being described. Various features may have changed in later releases, and, possibly, some features may have been removed.

---

## Version 12 Enhancements

Version 12 of CodeWatch complies with the RM/COBOL 12 release level.

**Note** CodeWatch version 12 will operate only with the version 12 RM/COBOL compiler and runtime system. However, RM/COBOL programs that are compiled with previous versions of the development system do not need to be recompiled in order to be used in CodeWatch. Both CodeWatch and the RM/COBOL version 12 development system will coexist with earlier versions on the same machine as long as the installation directories are different.

The other major enhancements in version 12 of CodeWatch include the following:

- CodeWatch now has the ability to debug service programs running under Xcentrinity Business Information Server (BIS) on an IIS web server. The IIS server can be local (that is, on the machine hosting CodeWatch), or remote and text files (for example, server response files) can be edited on the server from within the CodeWatch environment. For more information, see the *CodeWatch for Xcentrinity Business Information Server User's Guide Supplement*.

**Note** Remote compilation and deployment are not supported in this version.

- Filenames and pathnames in saved CodeWatch Workspace Files (that is, in **.cwf** files), are now stored as relative paths that are based on the directory that contains the saved workspace. As long as the files that make up the workspace can be located relative to the saved workspace file, this enhancement makes it possible to XCOPY a directory tree between workstations, check out a CodeWatch workspace from a version control system, or load a CodeWatch workspace over the network without adjusting pathnames.
- CodeWatch compatibility with Windows Vista and Windows Server 2008 has been improved. The RMCW.INI file is now stored in the

**%APPDATA%\RMCOBOL\CodeWatch** directory instead of the **C:\Windows** directory.

This release of CodeWatch also includes numerous defect corrections.

## Version 12.12 Enhancements

CodeWatch supports the new RM/COBOL 64-bit runtime system.

- If a 64-bit runtime is selected, it is run as a separate process by CodeWatch – specifically, in the DLLHOST.EXE surrogate host process. Note that the 32-bit runtime is executed in-process as before. Because of the different execution environment, there may be some minor differences in the way that 64-bit COBOL programs function under CodeWatch. One example: runtime-created Windows can appear behind the CodeWatch window. In some cases, it's possible that the program may have displayed a message box or may be waiting for an ACCEPT and that will cause CodeWatch to appear suspended. If, while debugging, you don't see the runtime window, please move the CodeWatch window or use Alt-Tab or the task bar to bring the runtime window to the front. Note that running CodeWatch on a second monitor avoids this situation completely and is highly recommended.
- When running on Windows 8 and later, 64-bit programs being debugged will be loaded into memory using large base addresses (that is, the program and data will reside above the 4GB line and have a base address that cannot be stored in 32-bits). If your program calls 3rd-party non-COBOL 64-bit programs, these will also be loaded at large base addresses. Because some 64-bit non-COBOL programs are converted 32-bit programs and may still store pointers in 32-bit integers, such 64-bit programs will not tolerate being loaded into the upper reaches of memory. This may cause them to fail when running under CodeWatch, but may run normally or fail intermittently when not running under CodeWatch. To debug such a 3rd-party program, start CodeWatch, run to the point where the 3rd-party program is loaded, and then attach Visual Studio to the DLLHOST.EXE process (not the CodeWatch RMCW.EXE process) and set breakpoints in the code.

This release of CodeWatch also includes numerous defect corrections.

## Version 12.14 Enhancements

- CodeWatch supports the new RM/COBOL 64-bit development system, including the 64-bit compiler.
- The RMCW.INI file is now stored in the **%APPDATA%\Micro Focus\RM\CodeWatch** directory instead of the **%APPDATA%\RMCOBOL\CodeWatch** directory. However, to maintain compatibility with previous versions, if an RMCW.INI file is found in the previous location, that file will continue to be used and updated.

This release of CodeWatch also includes numerous defect corrections.

---

## Version 11 Enhancements

Version 11 of CodeWatch complies with the RM/COBOL 11 release level and contains fixes for numerous minor problems. This is the first version of the product to support use on the Microsoft Windows Vista operating system.

---

## Version 10 Enhancements

Version 10 of CodeWatch complies with the RM/COBOL 10 release level. This version of CodeWatch also contains the following enhancements:

- The CodeWatch source code editor now has the ability to show and hide COPY files. By default, COPY files are now hidden, making source code editing and navigation faster. A hidden COPY file is denoted by a [+] symbol in the right margin of the source edit window. The hidden COPY file text can be revealed by clicking on the [+] symbol, and a revealed COPY file block can be hidden by clicking on the [-] symbol in the margin. In addition, the F11 key toggles the block that contains the cursor, and the View menu and the editor popup menu contain menu picks to perform these operations, and reveal and/or hide all COPY file blocks in the current file.

Note that Find command on the Edit menu will search only visible COPY file blocks. To search all COPY blocks, use the View | Copy Files | Expand All Blocks command.

- Double-clicking in the leftmost column now sets or clears a code breakpoint on that line.
- The File menu contains options to print the current source. The entire source or simply a selected portion of the source can be printed.
- CodeWatch displays a tab containing the name of each open file or window at the bottom of the document area. Clicking on the tab for a particular window brings that window to the top. This is called “workbook” mode, because the document area has the appearance of a binder. The previous mode can be selected with the "Settings" tab in the View | Preferences dialog box. This feature may also be referred to as “tabbed edit windows.”
- CodeWatch can load large projects and save large source files much faster than in previous versions.

This release of CodeWatch also includes fixes for numerous minor problems such as the problem of the system crash caused by double-clicking the Build button, and the failure to restore watchpoints with subscripts when loading a saved workspace.

---

## Version 9 Enhancements

Version 9 of CodeWatch complies with the RM/COBOL 9 release level and contains fixes for numerous minor problems. This version of CodeWatch also contains the following enhancements:

- Dependency checking is more complete. COPY file timestamps are now checked and dependency checking is more efficient overall.
- The time required to load large workspaces—that is, workspaces that contain a very large number of source and COPY files has been greatly reduced.

- Improved file loading performance for very large source files, especially when those files contain a large number of COPY file lines. Compilation performance has been improved, also. A progress indicator is displayed on the status bar during long operations.
- Full support for the RM/COBOL compiler ANSI/OEM switches:
  1. Workspaces are now declared to use the OEM or ANSI character sets. This selection is a workspace property and affects the editor, compiler, and runtime. The default is the character set chosen when the version 9 development system or runtime system was last installed.
  2. Previous versions of CodeWatch made no assumptions about the set of characters used in source files, which could cause OEM character sets to be displayed incorrectly.
  3. CodeWatch recognizes and uses the new `/cs_ansi` and `/cs_oem` runtime command line options.
  4. For OEM workspaces, CodeWatch allows the “RM Euro Codepoint” to be specified. This is useful when choosing a font that does not contain the Euro character as the default font.

For more information, see [Character Sets](#) (on page 104).

- A new dialog box is available from the [Edit menu](#) (see page 74). The Character Map dialog box displays a grid of characters in the currently-selected character set and also indicates the characters used in the file that is currently being edited. This is useful for editing programs that contain non-ASCII characters.

This dialog box also allows the font face name and font size to be chosen. CodeWatch allows a font to be chosen for OEM mode and a separate font to be chosen for ANSI mode. In OEM workspaces, fonts that contain an OEM or default to the ANSI character set, may be used. In ANSI workspaces, only fonts that use the ANSI character set may be used. For a complete description of this dialog box, see [Choosing an Editor Font](#) (on page 107).

For further information, see [Character Sets](#) (on page 104).

- The runtime `-V` (verbose) and `-F` (fill character) command line options are now recognized.
- CodeWatch now uses Windows XP Visual Styles.
- Educational and Evaluation versions of the RM/COBOL development system are now supported.
- Numerous minor problems have been corrected (such as the problem of the system crash caused by double-clicking the Build button).

---

## Version 8 Enhancements

Version 8 of CodeWatch includes several defect corrections, and the product complies with the RM/COBOL 8 release level.

---

## Version 7.5 Enhancements

Version 7.5 of CodeWatch contains the following enhancements:

- The ability to edit and compile your programs without leaving CodeWatch.
  - It is now possible to see and correct errors and warnings in an external text editor.
  - CodeWatch provides complete support for all RM/COBOL compilation options.
- CodeWatch now contains a fully-integrated text editor (called the Source Code Editor), with the following capabilities:
  - Unlimited levels of Undo and Redo.
  - Search and replace, including regular expression support.
  - Ability to invoke an alternate text editor of your choice with just a keystroke. Many external editors can even position the cursor at the current line and column.



# Index

- (minus sign) 47

## \$

\$ (dollar sign) 112

## +

+ (plus sign) 47

## A

All caps, as a document convention 3  
 Angle brackets, in program names 24, 27, 48  
 Animate button 28, 78  
 Animate command (Run) 28, 78  
 Animating the program 28, 78  
 ANSI 8, 105, 138  
 Attach Configuration utility (rmattach) 132, 133

## B

Bold type, as a document convention 3  
 Break command (Run) 78  
 Breakpoint command (Run) 78, 79  
 Breakpoints  
   changing 52, 88  
   deleting 52, 88  
   disabling 52  
   displaying (viewing) 52, 76  
   enabling 52  
   setting 29, 88  
 Breakpoints & Watchpoints dialog box 87  
   Code Breakpoints page 29  
   Code Breakpoints tab 88  
   Data Watchpoints page 31  
   Data Watchpoints tab 89  
 Breakpoints command (Edit) 75  
 Breakpoints Tab command (View) 76

Breakpoints tab, Data window 52  
 Build menu 76, 125  
 Build Options property page, Item Properties dialog box 126  
 Build Options property tab, Item Properties dialog box 93  
 Build Results Tab command (View) 76  
 Build Results tab, Data window 56, 130

## C

Call Stack Tab command (View) 76  
 Call Stack tab, Data window 55  
 Canceling a selection 118  
 Character Map dialog box 75, 101  
 Character sets 8, 39, 75, 83, 101, 104, 112, 138  
 Clear All command (Run | Breakpoint) 79  
 Clear command (Run | Breakpoint) 79  
 Clipboard  
   cutting, copying, and pasting text onto 123  
   using with selections 123  
   with the Edit menu commands (Source Code/Edit window) 121, 123  
 Close button 61  
 Close command 61, 69  
 Code *See* Source code  
 Code Breakpoints page, Breakpoints & Watchpoints dialog box 29  
 Code Breakpoints tab, Breakpoints & Watchpoints dialog box 88  
 code points 104  
 codepage 104  
 CodeWatch  
   animating the program 28, 78  
   changing the workspace 33  
   character sets 8  
   compiling 76, 93, 125  
   configuration 131  
   configuring a workspace 20, 35  
   configuring initialization file (rmcw.ini) 131  
   enhancements 1, 135  
   features, new 135, 137, 138  
   inspecting and modifying data 31, 66  
   installation 7  
   license, runtime system 132  
   main window, described 43  
   overview 11  
   rmcw.ini file, configuring 131  
   running the program 25  
   saving a workspace 33  
   setting  
     breakpoints 29, 52, 75, 76, 88  
     watchpoints 30, 53, 75, 76, 89  
   starting 17  
   stopping the program 32  
   system considerations 7

- system requirements, installation 7
  - tutorial, quick start 11
  - viewing and editing program source code 24, 97
  - workspace properties, setting 80
  - Colors, use of
    - in breakpoints 29, 52, 89
    - in code 32, 56, 67, 98
    - in watchpoints 53, 91
  - Command tab, Workspace Properties dialog box 87
  - Compilation options, setting 93, 126, 127
  - Compile All command (Build) 77, 125
  - Compile Changes command (Build) 125
  - Compile File command (Build) 76, 125
  - Compiler registry key, configuring 132
  - Compiling 125
    - Build menu commands 76
    - Build Results tab, Data window 56
    - changed files 125
    - compilation options, setting 93, 126, 127
    - environment variables 128, *See also* Environment variables
    - locating the RM/COBOL compiler 129
    - registering the RM/COBOL compiler 129
  - Configuration 131
    - compiler options 132
    - miscellaneous 134
    - runtime options 131
  - Context menus
    - Data window 57
    - Log window 58
    - Source Code/Edit window 99
    - Workspace window 49
  - Conventions and symbols 3
  - Copy command (Edit) 75
    - Source Code/Edit window 123
  - Copy files 49
  - Customer Care 5
  - Cut command (Edit) 74
    - Source Code/Edit window 123
- D**
- Data files
    - inserting 36, 71
    - type 55
    - viewing on Data Files tab 54
  - Data Files Tab command (View) 76
  - Data Files tab, Data window 54
  - Data Item command (View) 66, 76
  - Data items
    - inspecting 31, 66
    - pointer 67, 91
    - type 32, 53, 67, 91
  - Data Watchpoints page, Breakpoints & Watchpoints dialog box 31
  - Data Watchpoints tab, Breakpoints & Watchpoints dialog box 89
  - Data window
    - Breakpoints tab 52
    - Build Results tab 56
    - Call Stack tab 55
    - context-menu commands 57
    - Data Files tab 54
    - described 51
    - illustrated 27
    - Watchpoints tab 53
  - Debugging
    - Full Debug Information (Y) compilation option 24, 27, 97
  - Dependency checking 134
  - Directory search sequence 25, 85
  - Dockable windows 46
  - Docked command
    - Workspace windows 51, 57, 58
  - Document menu 60
  - Document windows 46, 59
    - closing 62
    - cycling through 61
    - menu 60
    - splitting 61
    - state of 59
  - Dollar sign (\$) 112
- E**
- Edit Item command (File) 73
  - Edit menu 74
    - Source Code/Edit window 123
  - Editing source code
    - with external text editor 110, 111, 114
    - with Source Code Editor 97
  - Editing techniques 115
    - correcting and editing text 121
    - cutting, copying, and pasting 121, 123
    - deleting text 121
    - extending selections 119
    - inserting new text 120
    - insertion point 116
    - moving around in a file 115
    - overwriting selected text 121
    - overwriting text 121
    - replacing text 121
    - selecting text 118
    - selections, editing operations with the Clipboard 123
      - using the Windows Clipboard 123
  - Ellipsis button, directory searches 25, 43, 87, 112
  - Enhancements to CodeWatch 1, 135
  - Environment tab, Workspace Properties dialog box 85
  - Environment variables

- PATH 23, 38, 42, 82, 86
- RMPATH 19, 128
- RUNPATH 19, 23, 38, 42, 82, 86
- Euro character 138
- Exit command (File) 74
- Expressions, regular 63, 64
- External text editor, configuring 110, 111, 114

**F**

- Features, new 135, 137, 138
- File menu 69
- Filenames, conventions for 3
- Files tab, Workspace window 49
- Find command (Edit) 63, 75
- Find dialog box 63
- Find Next command (Edit) 64, 75
- Find Previous command (Edit) 64, 75
- Floating command
  - Workspace windows 51, 57, 59
- Floating windows 46, 62
- Full Debug Information (Y) compilation option 24, 27, 97

**G**

- Glyphs 104

**H**

- Hyphens, optional, in RM/COBOL compilation and runtime options 4

**I**

- Icons, used in Workspace
  - window 47
- Initialization file (rmcw.ini), configuring 131
- Insert Items command (File) 71
- Insert mode 120
- Insertion point 116
- Inspect/Modify dialog box 31, 66
- Installation 7
- Installation and system considerations 7
- Introduction page, Workspace wizard 36
- Italic, as a document convention 3
- Item Properties command (File) 72
- Item Properties dialog box 91
  - Build Options property page 126
  - Build Options property tab 93
  - Programs property tab 93
  - Source File property tab 92

**K**

- Key combinations, document convention for 4

**Keyboard**

- basic techniques
  - correcting and editing text 121
  - deleting text 121
  - inserting text 120
  - moving the insertion point 117
  - selecting text 120

**Keys**

- for editing 121
- for moving the insertion point 117
- shortcut keys
  - on Build menu 76
  - on Edit menu 74
  - on Edit menu (Source Code/Edit window) 123
  - on File menu 69
  - on Run menu 77
  - on View menu 75
  - on Window menu 79

**L**

- Library files 20, 22, 42, 49, 86
- Licensed runtime system 132
- Locating source code files 25, 85, 97
- Log window
  - compiler output 130
  - configuring the font and point size 134
  - configuring the number of lines retained 134
  - context-menu commands 58
  - described 58
  - illustrated 27
- Log Window command (View) 76

**M**

- Main window, CodeWatch 43
- Maximize button 61
- Maximize command 61
- Menu bar and menus 44, 68
- Minimize button 61
- Minimize command 61
- Minus sign (-) 47
- Mouse
  - basic techniques
    - scrolling with 116
    - selecting text 118

**N**

- New command (File) 69

**O**

- OEM 105, 138
- Open command (File) 69
- Options command (Build) 77, 126

Options tab, Workspace Properties dialog box 83  
 Organization of this guide 2  
 Overtyping mode 121

## P

Paste command (Edit) 75  
   Source Code/Edit window 123  
 PATH environment variable 23, 38, 42, 82, 86  
 Plus sign (+) 47  
 Pointer data items 67, 91  
 Preferences dialog box 109  
   External Editing tab 110, 111, 114  
   Tabs tab 109  
 Program files 20, 49  
 Program Output command (View) 76  
 Program tab, Workspace Properties dialog box 81  
 Programs property tab, Item Properties dialog box 93  
 Programs tab, Workspace window 48  
 Properties  
   item 91  
   workspace 80

## R

Recent Workspaces command (File) 74  
 Redo command (Edit) 74  
 Registering the RM/COBOL compiler 129  
 Regular expressions 63, 64  
 Remove Item command (File) 72  
 Replace command (Edit) 65, 75  
 Replace dialog box 65  
 Restore button 60  
 Restore command 60  
 Results | Find Next command (Build) 77  
 Results | Find Previous command (Build) 77  
 Resume command (Run) 78  
 RM/COBOL  
   character sets 8  
   rmsetncls utility 8  
 RMPATH environment variable 19, 128  
 rmsetncls utility 8  
 Run | Breakpoint menu 79  
 Run menu 77  
 Running  
   animate the program 28, 78  
   control flow of execution 28  
   inspect and modify data 31, 66  
   modify and view watchpoints 30, 53, 89  
   program 25  
   Run | Breakpoint menu 79  
   Run menu 77  
   set breakpoints 29, 52, 88  
   stop program 32  
 RUNPATH environment variable 19, 23, 38, 42, 82, 86

Runtime Command line options 87  
 Runtime license 132  
 Runtime registry key, configuring 131  
 RuntimeLicensePathname option 132

## S

Save As command (File) 70  
 Save command (File) 70  
 Save Copy As command (File) 70  
 Saving a workspace 33  
 Scroll bars 116  
 Scrolling 116  
 Search string characters 63  
 Select All command (Edit) 75  
 Select Character Set dialog box 95  
 Selecting  
   with the keyboard 120  
   with the mouse 118  
 Selecting text 118  
 Selection mode 118  
 Set Additional Options page, Workspace wizard 22, 40  
 Set command (Run | Breakpoint) 79  
 Set Environment page, Workspace wizard 22, 41  
 Set Program to Run page, Workspace wizard 21, 38  
 Shortcut keys  
   on Build menu 76  
   on Edit menu 74  
   on Edit menu (Source Code/Edit window) 123  
   on File menu 69  
   on Run | Breakpoint menu 79  
   on Run menu 77  
   on View menu 75  
   on Window menu 79  
 Source code 49, *See also* Source Code Editor; Source Code/Edit window  
   compilation results on Build Results tab 56  
   compiling 125  
   editing 24, 97  
   locating 97  
   viewing 24, 103  
 Source Code Editor 97, *See also* Source code; Source Code/Edit window  
 Source Code/Edit window *See also* Source code; Source Code Editor  
   basic editing techniques 115  
   correcting and editing text 121  
   cutting, copying, and pasting 121, 123  
   deleting text 121  
   extending selections 119  
   inserting new text 120  
   insertion point 116  
   moving around in a file 115  
   overwriting selected text 121  
   overwriting text 121

- replacing text 121
- selecting text 118
- selections, editing operations with the Clipboard 123
- using the Clipboard 123
- context-menu commands 99
- creating new source code file 101
- editing source code 24, 97, 103
- illustrated 27, 98
- locating source code files 85, 97
- opening existing source code file 102
- preferences, setting 109
- viewing source code 24, 97, 103
- Source File property tab, Item Properties dialog box 92
- Source files *See* Source code; Source Code Editor; Source Code/Edit window
- Special characters, regular expressions 64
- Splitting windows 61
- Start command (Run) 78
- Starting CodeWatch
  - from a DOS Command Prompt 18
  - from a saved workspace 17
  - from RM/COBOL Runtime Command line 19
  - from Windows 17
- State | Disable All command (Run | Breakpoint) 79
- State | Disable command (Run | Breakpoint) 79
- State | Enable All command (Run | Breakpoint) 79
- State | Enable command (Run | Breakpoint) 79
- State | Toggle command (Run | Breakpoint) 79
- Status bar 45
- Status Bar command (View) 75
- Step Paragraph command (Run) 78
- Step Program command (Run) 78
- Step Section command (Run) 78
- Step Statement command (Run) 78
- Stop command (Run) 78
- Stopping the program 32
- Support services, technical 5
- Symbols and conventions 3
- Synonyms 23, 42, 86
- System considerations 7
- System installation 7
- System requirements, installation 7

## T

- Tab characters 109
- Text patterns 64
- Title bar 44
- To Cursor command (Run) 78
- To Program Change command (Run) 78
- Toggle command (Run | Breakpoint) 79
- Toolbar 45, 68
- Toolbar command (View) 75, 76
- Tutorial 11

## U

- Undo command (Edit) 74, 79

## V

- View menu 75

## W

- Watchpoints
  - changing 89
  - deleting 89
  - displaying 53, 76
  - setting 30, 89
- Watchpoints command (Edit) 75
- Watchpoints Tab command (View) 76
- Watchpoints tab, Data window 53
- Window menu 79
- Windows
  - dockable 46
  - document 46, 59
  - floating 46, 62
- Workspace
  - changing 33
  - commands (File) 70
  - configuring 20, 35
  - defined 21, 35
  - filename extension 17, 33
  - properties, setting 80
  - saving 33
  - starting CodeWatch from 17
  - window
    - context-menu commands 49
    - described 47
    - icons used 47
    - illustrated 23, 27
    - wizard 11, 20, 35
- Workspace | Close command (File) 70
- Workspace | New command (File) 70
- Workspace | Open command (File) 70
- Workspace | Properties command (File) 71
- Workspace | Save As command (File) 71
- Workspace | Save command (File) 71
- Workspace Properties dialog box 80
  - Command tab 87
  - Environment tab 85
  - Options tab 83
  - Program tab 81
- Workspace window
  - context-menu commands 49
  - described 47
  - Files tab 49
  - icons used 47
  - illustrated 23, 27
  - Programs tab 48

- types of
  - dockable 46
  - document 46, 59
  - floating 46, 62
- Workspace wizard 11, 20, 35
  - Introduction page 36
  - Set Additional Options page 40
  - Set Environment page 41
  - Set Program to Run page 38