

# **Getting Started Guide**



**Version 8**

© 1996, 1998



1830 West University Drive, Suite 112 Tempe, AZ 85281 USA  
Voice (602) 968-1945 Fax (602) 966-1654  
Email: [tech@multiedit.com](mailto:tech@multiedit.com) WWW: <http://www.multiedit.com>

## ***SINGLE-USER LICENSE AGREEMENT***

---

Multi-Edit is protected under United States copyright law. Copies may be made for the purpose of backup ONLY. If this software was purchased as a single user copy, the software may be moved from one computer to another and may be used by more than one person. However, THIS SOFTWARE MAY NOT BE USED ON MORE THAN ONE COMPUTER, OR BY MORE THAN ONE PERSON, AT THE SAME TIME unless covered under a separate License Agreement with American Cybernetics, Inc., the terms of which Agreement supersede this Single-User License Agreement. To obtain a multiple-user site or network license, contact American Cybernetics, Inc.

## ***LIMITED WARRANTY***

---

American Cybernetics warrants the software media in this package against defects in materials or workmanship for a period of thirty days. If, in this period, the software media is found to be defective, it may be returned to us for free replacement.

American Cybernetics makes no representations or warranties as to the merchantability or fitness of this product to a particular purpose. There is no other warranty, express or implied.

## ***TECHNICAL SUPPORT***

---

Unlimited free technical support can be received through fax or the Internet. By registering your copy of Multi-Edit, you are entitled to free phone support.

On the Internet, our technical support email address is [tech@multiedit.com](mailto:tech@multiedit.com). Our web site URL is <http://www.multiedit.com>. In addition, users can download upgrade patches and new information about Multi-Edit at <ftp://ftp.multiedit.com>. Log in as *anonymous* and use your complete email address as a password.

Technical support is available via fax at (602) 966-1654. Be sure to provide your name, serial number, detailed description of the problem, and a return fax number.

Registered users may call (602) 968-1945 to receive phone support. Please have your serial number ready when you call.

© 1996, 1998 by American Cybernetics, Inc. All rights reserved. Trademarks on all Multi-Edit products are the property of American Cybernetics, Inc. Other trademarks contained herein are the property of their respective holders.

# Table of Contents

---

SINGLE-USER LICENSE AGREEMENT .....	2
LIMITED WARRANTY .....	2
TECHNICAL SUPPORT .....	2

<b>Table of Contents .....</b>	<b>3</b>
--------------------------------	----------

<b>Introduction .....</b>	<b>5</b>
---------------------------	----------

Operational Overview .....	5
Software Requirements .....	5
Hardware Requirements .....	5
Package Contents .....	6
Network Version .....	6
Key Assignments .....	6
Installation .....	6
Setup Options .....	7

<b>Using Support Resources .....</b>	<b>9</b>
--------------------------------------	----------

The Getting Started Guide .....	9
The Help System .....	9
Technical Support .....	10

<b>The Multi-Edit Interface .....</b>	<b>13</b>
---------------------------------------	-----------

Command Sets .....	13
Keys/Commands .....	15
Menu System .....	17
Toolbar/Toolbox .....	20
Using the Mouse .....	23
Tab Bar .....	24
Window List Box .....	26
Editing Windows .....	27
Results Window .....	30
Status Bar .....	31
List Boxes .....	32



**Getting Started Using Multi-Edit..... 33**

- Opening Files ..... 33
- Viewing Files ..... 35
- Editing Text ..... 37
- Formatting Text ..... 42
- Templates ..... 44
- Search Operations ..... 47
- Collapsing Text ..... 53
- Multi-Tags ..... 54
- Project Management ..... 55
- Session Management ..... 56
- Compiling Within Multi-Edit ..... 57
- Comparing Files ..... 59
- Multi-Edit Macros ..... 61
- Version Control Support ..... 62

**Additional Features..... 67**

- Help System Management ..... 67
- Language Support in Multi-Edit ..... 68
- Borland IDE Integration ..... 74
- Programming for the Internet ..... 75
- Startup Options ..... 80
- Command Line Options ..... 80
- Additional Tools ..... 81

**Index..... 85**

# Introduction

---

Congratulations! You have made an excellent choice. Multi-Edit 8 is the most powerful and easy-to-use 32-bit text editor available today. You will find that it will serve your needs long into the future, since we at American Cybernetics continually strive to keep our products on the forefront of current technology.

---

*As a registered Multi-Edit user, you will always have access to the latest updates and add-ons to expand and enhance your editing environment. Please remember to register via mail, fax, email or our website.*

---

## **OPERATIONAL OVERVIEW**

---

Multi-Edit is a programmer's text editor with powerful features designed to deliver the ease of operation and timesaving functions you need to meet demanding deadlines.

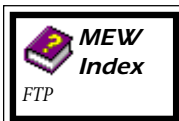
With Multi-Edit, you can manipulate text with unsurpassed ease and compile source files while you are still in Multi-Edit. Your ability to handle files is greatly enhanced. Up to 256 files can be edited simultaneously. Multi-Edit 8 effortlessly handles large files, too. Line lengths up to 16K are no problem.

You will not spend a lot of time getting up to speed with Multi-Edit either. You will be doing productive work very quickly with the aid of our intuitive user interface. Drop down menus and special Key Assignments help you execute commands in a heartbeat. Plus, features like our smart indent, template editing, and construct matching make our language support second to none. And Multi-Edit's Help System and Technical Support guarantees you will never encounter a problem without a solution near at hand.



## **SOFTWARE REQUIREMENTS**

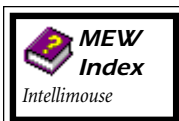
---



Multi-Edit 8 requires Microsoft Windows 95/NT 3.51 or greater to run. To use FTP features in Multi-Edit, you must have Microsoft Internet Explorer 3.0 or greater (or Wininet extensions) installed.

## **HARDWARE REQUIREMENTS**

---



Multi-Edit 8 can run on an IBM-compatible 486 or Pentium computer with a minimum of four MB of available memory. Disk space requirements are approximately six MB (assuming you install everything). Intellimouse support requires the Microsoft Intellimouse.

## **PACKAGE CONTENTS**

---

Multi-Edit 8 is shipped on 3.5" 1.44 MB diskettes or CD-ROM. A registration card, *Getting Started Guide*, and product literature are also provided.

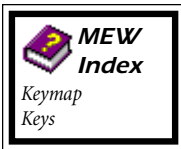
## **NETWORK VERSION**

---

Network users can benefit from Multi-Edit's many capabilities with our Network version. All information in the *Getting Started Guide* and Help System is applicable to network users. If you have purchased the network version, please refer to the *Network Guide* included with your package. It contains instructions on installation and functionality specific to the network version.

## **KEY ASSIGNMENTS**

---



As installed, the default keymap is the CUA-style keymap. Additional keymaps for use within Multi-Edit 8 are Brief, WordStar, Borland IDE, and Visual Studio IDE. In light of all these choices and the complete configurability of the key assignments, keep in mind that key assignments referred to in the *Getting Started Guide* and the Help System reference the CUA keymap and may not match your current key assignments. However, the Menu System will reflect the keymap you are currently using.

## **INSTALLATION**

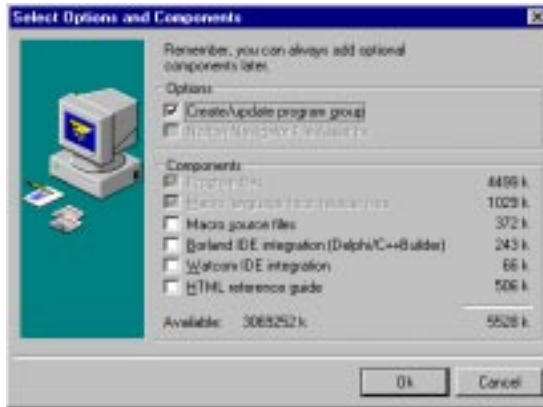
---

- ▼ Make a backup copy of your master diskettes (Multi-Edit is not copy-protected). Write the serial number on the inside cover of this *Guide*. You will need to reference this number during the installation. You will also need to have your Multi-Edit for Windows 7 serial number available if you are upgrading (or the Multi-Edit DOS serial number for crossgrade). Be sure to register Multi-Edit Version 8 so that you can obtain technical support and receive information about future upgrades and patches.
- ▼ Insert diskette 1 or the CD-ROM.
- ▼ From the Windows Control Panel, select **Add New Programs | Install** and follow the on-screen instructions. Or, from the Windows **Start Menu | Run**, type A:\MEWSETUP.EXE in the **Open** text box (use the appropriate drive letter) and select **OK**.
- ▼ Select the appropriate options at the introductory installation screen. **DO NOT** select **Add options to existing install** unless you have already installed *this* version of Multi-Edit and want to add options (such as source code, help files, Delphi integration, etc.) to your install.

- ▼ Follow the on-screen instructions to install Multi-Edit and/or the optional components. *Setup Options* are described in the next section.
- ▼ When prompted, provide the serial number and press <Enter>.
- ▼ Once installed, Multi-Edit will start and load a README.TXT file that contains last minute information and release notes that you should read.

## SETUP OPTIONS

Several options are included in the installation of Multi-Edit 8.



- ▼ **Program files**—This option must be selected to install necessary program files for Multi-Edit to run.
- ▼ **Macro Language Help**—Select this option to install the related Help files for Multi-Edit's CMACW Macro Language.
- ▼ **Macro Source Files**—This option will install the source for the system macros. This is only needed if you are planning on modifying the system behaviors not available in setup.
- ▼ **Borland IDE Integration**—Adds transparent syncing for Delphi and C++ Builder. You will be asked to supply the path in which the binary directory is located for each program.
- ▼ **Watcom IDE Integration**—Provides all hooks into Watcom IDEs. Supports 10-11 C and Fortran.
- ▼ **HTML Reference Guide**—This handy on-line Help System provides a reference for HTML tags.



# Using Support Resources

---

We want you to be able to easily find what you need to most effectively use Multi-Edit 8 as your text editor. There is information in this *Getting Started Guide*, the online Help System, as well as the American Cybernetics web site and FTP site. Our technical support representatives are available via email, fax and telephone.

## ***THE GETTING STARTED GUIDE***

---

The purpose of this *Getting Started Guide* is to assist you with beginning to use Multi-Edit 8. This *Guide* does not contain *all* of the reference material you will need, as we have opted to replace the manual and reference with an improved online Help System. This allows us to continue to offer Multi-Edit at a competitive price while providing current software documentation with each release.

In this Guide, you will find illustrated information on *The Multi-Edit Interface* so that you will feel more comfortable getting around. Next, we will introduce you to key concepts for *Getting Started Using Multi-Edit*. Finally, we review some *Additional Features* that you may want to explore on your own, using the Help System for guidance.

While the information included in this *Getting Started Guide* was accurate at the time of printing, you can expect to find current details on changes and improvements to Multi-Edit 8 to be reflected in the on-line Help System, which is continually being updated. Please visit our website to download updates to the Help System as they become available.

## ***THE HELP SYSTEM***

---

The Help System will be your primary source of information after you have reviewed this *Guide*.

In addition to the Multi-Edit Help file (MEW32.HLP), there are three additional help files available: CMACW Macro Language Reference (CMACWIN.HLP), Dynamic Data Exchange Help (MEWDDE32.HLP), and WebLair HTML Reference (WLHTML95.HLP). All of these files can be accessed from the **Help Menu**. In addition, toolbar buttons on the main help windows allow you to quickly access any of the other three help files. We have used comprehensive indexing in our Help System, to ease your search for information using the Contents, Index and Find tabs.



Help System Keywords (shown at left) will be used throughout this *Getting Started Guide* when a topic is covered in more detail in the Help System. The first line shows which Help file to search (MEW, CMAC, DDE, or HTML). The second line shows which portion of the Help file to search (Contents, Index, or Find tabs). The remainder shows which keywords can be used to heirarchically search the Help file for the given topic.

The Help System files are actively being updated, along with improvements to Multi-Edit 8. Be sure to check periodically for the updated Help files that are available for download from the American Cybernetics website.

## ***TECHNICAL SUPPORT***



If you cannot find the answer you need in this Guide, the Help System, or on our web site, you may contact Technical Support via email ([tech@multiedit.com](mailto:tech@multiedit.com)), fax (602-966-1654) or phone (602-968-1945). Please supply your serial number and version information (found under **Help | About**) when contacting Technical Support.

### ***American Cybernetics on the Internet***

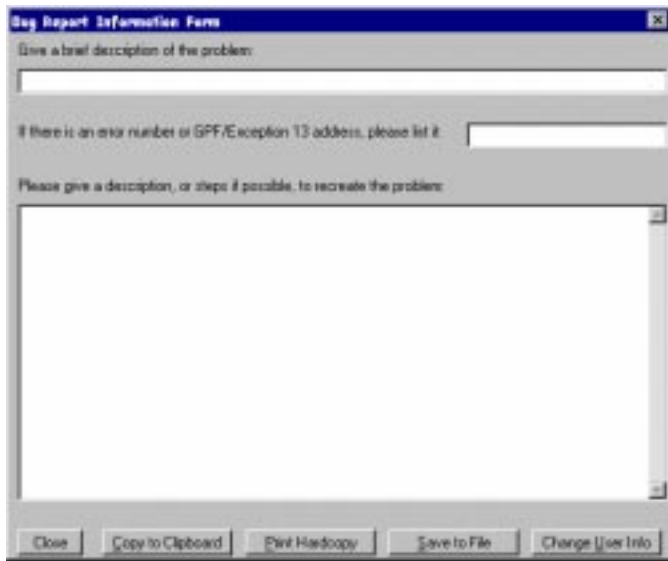
Stay current with Multi-Edit developments by visiting the American Cybernetics web site at <http://www.multiedit.com>. There you will find product information, FAQs, technical articles, updates (to the program and the Help system), patches, and links to our ftp site ([ftp.multiedit.com](ftp://www.multiedit.com)).

### ***Email to Technical Support***

When sending email to Technical Support, please use **Help | Create Bug Report** to assist us with solving your problem. The Bug Report User Info need only be filled out once, but can be modified later.

The image shows a Windows-style dialog box titled "Bug Report User Info". It contains several input fields and checkboxes. The "Name" and "Email" fields are empty. Under "Hardware Information", "Machine Mfg:" is "Dell, Zecc, Gateway, IBM, ect...", "CPU type and spd:" is "486-dc2/66", "Video Card:" is "ATI Mach 32, Viper VLB, etc...", "Memory Avail:" is "In Megabyte", and "Network Type:" is "Novell, Lantastic, etc...". Under "Software Information", "Operating System:" is "OS/2 ver 3(Warp), Windows 95, Wind", "MEW Version:" is "1.0 Beta 5a", "MEW Install Dir:" is "C:\MEW8\", and "Command Map:" is "CMDMAP". There are two checkboxes: "MEW Install Dir is a Network Drive" (unchecked) and "Command Map Mode" (checked). "OK" and "Cancel" buttons are at the bottom.

Please fill out the form as completely as possible and press **OK** to display the Bug Report Information Form:



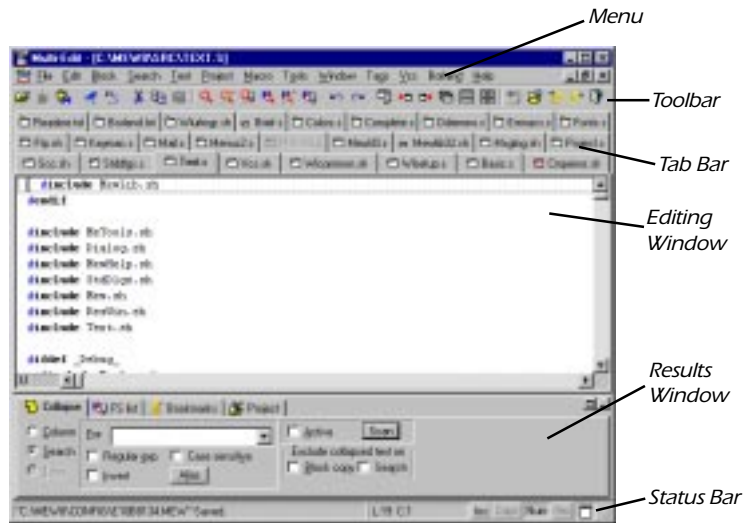
The screenshot shows a dialog box titled "Bug Report Information Form" with a close button (X) in the top right corner. The form contains three input fields and a large text area. The first field is labeled "Give a brief description of the problem:". The second field is labeled "If there is an error number or GPF/Exception T3 address, please list it:". The third field is labeled "Please give a description, or steps if possible, to recreate the problem:". At the bottom of the dialog box, there are five buttons: "Close", "Copy to Clipboard", "Print Hardcopy", "Save to File", and "Change User Info".

Press **Copy to Clipboard**, then paste the Bug Report in an email message to Multi-Edit Technical Support. Press **Print Hardcopy** to print the Bug Report and fax it. Press **Save to file** to save the Bug Report to a file that can be sent as an email attachment.



# The Multi-Edit Interface

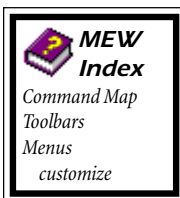
Multi-Edit 8 features full 32-bit performance with an improved interface including tabular window selectors and dialog boxes, automatic window sorting, a tabular Results Window that displays search results and more. This section describes the basics of the Multi-Edit screen and navigation.



## COMMAND SETS

The following information is stored within a single Command Set under Multi-Edit:

- ▼ Key/Command Mapping
- ▼ Toolbars/Boxes
- ▼ Menus

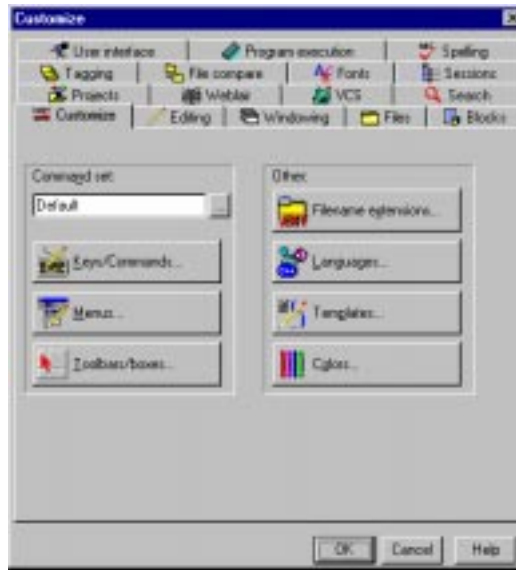


When you load a Command Set, you load *all* of the above components for the Command Set. If you create a new Command Set you will have to create Keys/Commands, Toolbars and Menus for the Command Set. Therefore, we suggest that you copy an existing Command Set and modify it, rather than create your own completely from scratch.

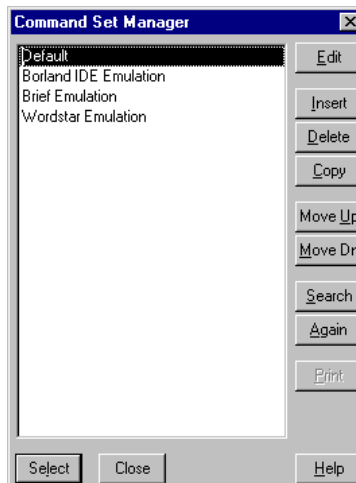
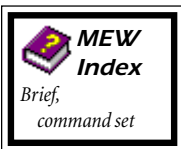
*NOTE: A Command Set is more than just keys; it contains menus and toolbars*

To view the Command Set Manager dialog,

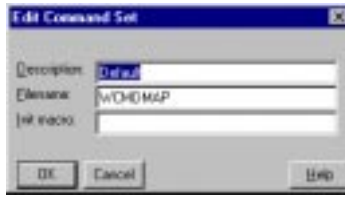
1. Select **Customize** from the **Tools Menu** and click on the **Customize** tab.



2. Press the ... button to the right of the text box below **Command set**. This will display a standard Multi-Edit list box with a list of existing Command Sets. This dialog box allows you to create, manage and select whole key/command mappings while keeping those command maps separate from each other. It is here that you can select some of the optional keymappings, such as the Brief keymap.



- Press the **Select** button to enable the currently highlighted command mapping. Select an item in the list and press the **Edit** or **Insert** button to display the Edit Command Set dialog box.



## KEYS/COMMANDS



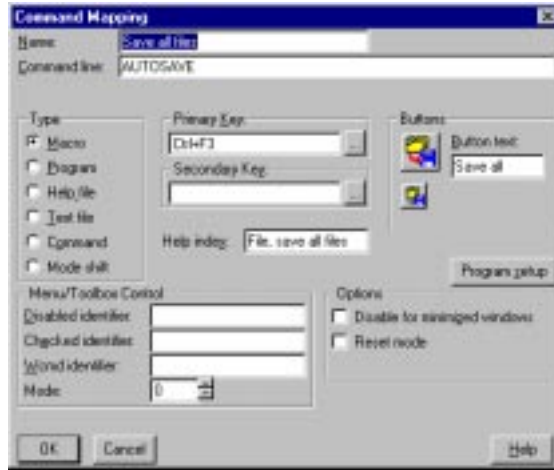
**Tools | Customize | Keys/Commands** allows you to change almost every key assignment in Multi-Edit. We have included several examples to help you set up, modify and customize the key/command map to your liking.



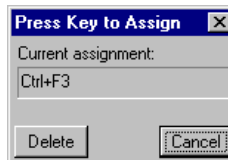
Each keymap line displays a command description or macro description, toolbar icons associated with the command, button text, and the Key Assignments for invoking that command or macro. Many of the lines have two Key Assignments. Multi-Edit allows you to assign a primary key and an alternate key to a command or macro. Only the first will show if a menu item is associated with the command.

To delete a key assignment or change a current key assignment to a different key, follow these steps:

1. Find the key mapping entry that you want to change using the search feature in the dialog, or by browsing through the list. Once you have found the entry, highlight it and press the **Edit** button.



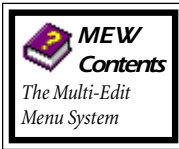
2. The Command Mapping dialog is shown. You will be primarily interested in the **Primary Key** and **Secondary Key** fields. Select the browse button to the right of the **Primary Key** field.



3. The Press Key to Assign dialog appears. If you want to remove this key assignment (thus freeing the key up for something else you had in mind), press the **Delete** button. Otherwise, press the new key(s) you want assigned to this command.
4. Repeat steps 2 and 3 for the **Secondary Key** field, if desired.
5. Press **OK, Close, OK** in the successive dialog boxes.

## MENU SYSTEM

Multi-Edit's menu system was designed to be easy to use, intuitive, and fully configurable.



File Edit Block Search Text Project Macro Tools Window Tags Vcs Borland Help

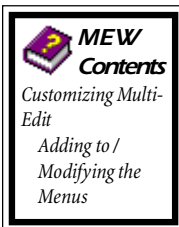
Most references to the menus in this *Guide* are specific to the default menus. Menu items, menu options, and dialog field names are displayed in bold, separated by the | character when you are to make a series of selections. For example, **Tools | Customize | Menu | Main**, refers to the **Tools Menu**, **Customize** option, **Menu** button, **Main** field.

Some menu options have corresponding keyboard shortcuts. For example, in the default configuration, **File | Open** can also be performed by pressing the **F3** key. Menu and keyboard mapping can be user-configured. Because of this, and to reduce confusion, default keymapping will not be noted in this *Guide*.

### Customizing Menus

Adding items to the menu is a snap when you follow these easy steps:

1. Bring up the Menu dialog by selecting **Tools | Customize | Menu**.



2. Select the menu you want to modify and press **Edit** to display the current layout of the selected menu.

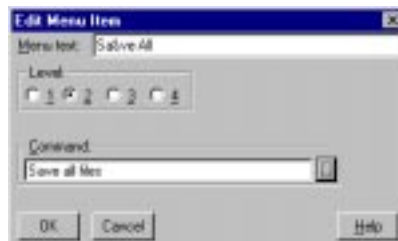


3. Select a menu item and press the **Edit** button, to display the Edit Menu set dialog.

**Menu Text**—Precede menu quick-keys with an ampersand (&) character. For example, if you want **Menu** to appear with 'M' being the quick-key, type **&Menu**.

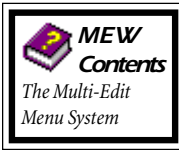
**Level**—This is the level of this particular menu item. This field controls whether the item is placed along the main menu, or as a sub-menu.

**Command**—This field contains a link to a specific command map entry. See below for more information on this field.



4. To set or change the command map entry linked to this menu item, press the ... button to the right of the **Command** field. The Command Mapping dialog will appear (discussed previously in the *Keys/Commands* section). Highlight the command you want to link to the menu item and press the **Select** button. If no command mapping item exists for this menu item, you may create one.
5. Press **OK**.

## Default Main Menu



The Default **Main Menu** belongs to the Default Command Set. In addition to the default menus, there may be a **Borland Menu** for Borland Integration and/or a **Watcom Menu** for Watcom Integration, depending on the Multi-Edit options you have installed.

### ***File Menu***

The **File Menu** provides options for managing, retrieving, printing, sending, and saving files. In addition, the command shell and Session Management can be accessed from the default **File Menu**.

### ***Edit Menu***

The **Edit Menu** offers convenient tools for cutting and pasting text to and from the clipboard, as well as **Undo**, **Redo**, and **Repeat**.

### ***Block Menu***

The **Block Menu** provides commands for marking blocks of text, and for moving and copying text you have marked.

### ***Search Menu***

The **Search Menu** provides commands that allow you to search for text, search and replace text, save and retrieve cursor positions, and move to a specific line number. All of the search dialogs have been merged into a single, tabbed dialog.

### ***Text Menu***

Through the **Text Menu**, you can alter the formatting of a document and manipulate text in a variety of ways.

### ***Project Menu***

The **Project Menu** provides access to the Project Manager commands, including FTP file transfers.

### ***Macro Menu***

The **Macro Menu** contains tools for executing and debugging Multi-Edit macros. In addition, keystroke macros may be edited from this menu.

### ***Tools Menu***

The **Tools Menu** is an eclectic assortment of tools, shortcuts, and utilities that help to supercharge your text editing and programming capability. **Tools | Customize** provides a wide assortment of configuration options for Multi-Edit 8. American Cybernetics has striven to provide the most user-configurable text editor available. We urge you to explore the myriad of configuration options Multi-Edit 8 offers.

## Window Menu

The Window menu provides a variety of commands for navigating between, sizing, and manipulating windows.

## Tags Menu / Multi-Tags

Multi-Tags is an easy-to-use, hypertext-like source code browser for C/C++, CMACW, Turbo-Pascal, Java, Fortran, ASM, Modula-2, dBase/xBase/Clipper and Paradox. Any text file may contain tags via explicit tags.

## Vcs Menu / Version Control Support

VCS stands for Version Control System. One of the main purposes of VCS programs is to help maintain and track versions or revisions of a set of files.

## TOOLBAR/TOOLBOX

Many common editing tasks have an icon associated with the command and a button set up in the toolbox for easy mouse access. By default, the Main Toolbar is positioned below the Main Menu. You can customize the toolbox to suit your editing needs by modifying, removing, or creating buttons and toolboxes. You can configure the toolboxes to be displayed along any window border, or even have them “float” in the Multi-Edit workspace.



### Configuring the Toolbar



While most of the toolbar configuration can be done from within the **Edit Toolbox** dialog, (**Tools | Customize | Toolbars/boxes**) a right mouse click on a toolbar or toolbox will display a pop-up menu with the following options:

- ▼ **Left vertical, Right vertical, Top horizontal, Bottom horizontal, Floating, Roving**—Only one of these items may be selected at a time. Toolbar/box positions may be along any border, “floating” in the Multi-Edit workspace, or “roving” on the Windows desktop.
- ▼ **Big icon, Small icon**—Controls whether the buttons displayed in the toolbar/box are large or small icons.
- ▼ **Hide toolbox**—If you do not want the toolbox to appear in the Multi-Edit workspace, select this item. The toolbox selected will be hidden from view. To make it reappear, uncheck the **Hidden** check box in the Edit Toolbox dialog.
- ▼ **Edit toolbox**—Brings up the Edit Toolbox dialog for that toolbar/box. From it, you may add, remove, or edit buttons in the toolbar.
- ▼ **Toolbox Manager**—Brings up the Toolboxes dialog.

To add to or modify items in the toolbars, follow these steps:

1. Bring up the Toolboxes dialog by selecting **Tools | Customize | Toolbars/boxes**.



2. Highlight the Toolbar and press the **Edit** or **Insert** or **Copy** button to modify or create a toolbar.



**Title**—This text box contains the name of the toolbox you are editing.

**Hidden**—If checked, then the toolbox will not be active. In order to use the toolbox, this must be unchecked.

**Position**—This drop-down list box allows you to configure where the toolbox will appear in Multi-Edit. There are six options available: along one of the borders, floating, or roving. Floating toolboxes can be positioned within the Multi-Edit window; roving toolboxes can be positioned anywhere on the Windows desktop. They may sized with multiple rows and columns.

If multiple toolbars are positioned in the same location, the order in which the Toolbars will be displayed is based on the order in the Toolboxes dialog.

**Style**—This list box lets you choose between large or small icons in your toolbox.

**Show On**—This field designates a global variable or macro used to instruct Multi-Edit when to hide or show the toolbar. If it refers to a global variable, the variable name is entered here. If it refers to a macro, the macro name must be preceded by an equals sign (=), and the macro must be a string function with no parameters (C-style). The variable or macro must return 1 (TRUE) if the toolbar is to be hidden. /LS=lang may also be used, where "lang" is the name of the language that the toolbar is to be SHOWN. For example, entering /LS=DELPHI would tell it that the toolbar is only to be shown when a Delphi file is being edited. The /LS= syntax has also been added to the "Disabled identifier" in Command Map Editing.

3. Give your toolbar a name (if you are creating one), ensure its hidden/unhidden state is set correctly, and set the icon position and style.
4. Pressing **Insert** or **Edit** will cause the Command Mapping dialog will appear (discussed previously in the *Keys/Commands* section). Highlight (or press **Insert**, if you want to add a new command) the command you want to put into the toolbar and press **Select**.
6. To organize items in the toolbar, use the **Move Up/Move Down** buttons.

### Using Custom Buttons on the Toolbars



In order to add custom buttons to your Multi-Edit toolbars, you must have access to a resource editor—you cannot use Multi-Edit to edit your toolbar buttons. Such resource editors are included with many of the popular IDEs (especially for C) or are available separately. Whatever application you use to edit your bitmaps, it must be able to save to a .DLL file, which is where Multi-Edit accesses the toolbar bitmaps.

---

*NOTE: The toolbar images in Multi-Edit are not icons in the strict sense—they are bitmaps.*

---

Included with Multi-Edit is a USERBMP32.DLL file, where you can edit and add your user bitmaps. Use your resource editor to open USERBMP32.DLL and add your bitmaps. So your bitmaps match the current MEW standard, large buttons are 25x25 pixels, and small buttons are 15x15 pixels. Use the resource editor to create or copy bitmap images into the USERBMP32.DLL file. Take note of the names of the resources you wish to incorporate in Multi-Edit (for example: USER\_100).

Once you have your custom images within the USERBMP32.DLL, start up Multi-Edit and enter the **Tools | Customize | Keys/Commands** dialog. Highlight and edit the item you wish to use your custom button for, then press the large or small button in the **Buttons** field (whichever one you would like to set). A list of available bitmaps will appear. Press the **Insert** button. When prompted for the resource name, enter in the name you noted above (for example: USER\_100). The new bitmap will be added to the list. You may then highlight and **Select** that bitmap for use by your command mapping.

---

## **USING THE MOUSE**

---

We have added improved functionality to the mouse in Multi-Edit 8, including Microsoft Intellimouse support, right-click pop-up Context Menus, as well as drag and drop for files and blocks.

### **Intellimouse Support**



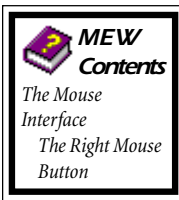
Scrolling in Editing Windows works with the Microsoft Intellimouse. The wheel will scroll editing windows and list boxes up or down one line at a time. Hold down the <Ctrl> key to scroll one page at a time.

---

*Multi-Edit 8 supports the standard wheel messages that have been built into Windows. Intellimouse support in Multi-Edit may work with other wheeled mice, but only the Intellimouse has been tested.*

---

### **Right Mouse Button**



The right mouse button has special functionality in Multi-Edit 8. During normal editing, you can click the right mouse button in various areas of the screen to access a pop-up **Context Menu**.

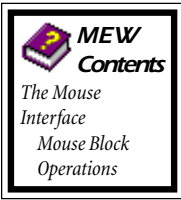
Exactly which menu gets accessed depends on where the mouse cursor is when clicked. Right mouse button functionality for the Editing Windows, toolbars, the Tab Bar, the Results Window and the Status bar is described in more detail throughout this *Guide* and the Help System.

### **Drag and Drop Files**



From the Windows Explorer, you can open a file by selecting the file you want to open with the mouse and dragging it to the Multi-Edit screen. If you have minimized Multi-Edit, you may also drag the file to the minimized icon. When using the drag and drop feature, Multi-Edit opens a new window for each file loaded. Files are opened in the default style for its extension.

## Moving and Copying Blocks



If you click and hold the left mouse button on any position inside a marked block, you can drag the mouse cursor to the position to which you want to move the block. You will notice a small icon displayed next to the mouse cursor, indicating that a block move operation is occurring. When you release the mouse button, the block will be moved to that position. While marking, right-clicking while holding the left mouse button down will change the mouse mode.

To copy a marked block to another location without removing it from the first location, hold the <Alt> key down while keeping the left mouse button pressed. Move the mouse cursor to the desired position and release both buttons.

## TAB BAR



The Tab Bar, a tabular window selector, appears by default across the top of the Multi-Edit window under the toolbars. The Tab Bar can be placed above or below the Editing Windows. Click on a tab in the Tab Bar to bring the selected Editing Window to the top.



Bitmaps on the tabs show the status of a file:



File has been modified.



File is read only.



Window is minimized.



Window is hidden (grayed title if Show Hidden has been enabled).

The Tab Bar is configurable via the **Tools | Customize | Windowing** dialog.



- ▼ When enough windows are open such that the Tab Bar gets filled on one line, remaining tabs can be displayed on additional lines or a single line (arrow buttons will be displayed on the right that can be clicked to move through the lines of tabs) depending on the configuration
- ▼ The Tab Bar will initially be displayed with the file names in tabs. However, when the Tab Bar expands to multiple lines, clicking on a tab brings it to the bottom and the order in which the tabs are displayed changes. To avoid this, you can customize the Tab Bar to display files in “Button” style, similar to the Windows Task Bar. Changes to the “Button” style option will not affect tabbed dialogs.



- ▼ File names can be displayed in Proper Case, which saves space but does not reflect true file name case.
- ▼ Normally, windows that are hidden or minimized are not shown on the Tab Bar. This can be changed by customizing the windowing (**Tools | Customize | Windowing**).

---

*NOTE: Fonts, font colors, and tab colors can be customized as well. Changes to font and tab colors will affect all tabbed dialogs.*

---

### Right Click Options

Right click on a tab in the Tab Bar to display a pop-up **Context Menu** which includes the following:

- ▼ **Window Functions**—Restore, Move, Size, Minimize, Maximize, Close, Hide, Previous, Next
- ▼ **Extension setup**—Almost everything in Multi-Edit is filename extension driven. This allows you to have one setup for your .PAS files, and a completely different setup for your .CPP files. Filename specific setup can also be configured under **Tools | Customize | Filename Extensions**.
- ▼ **Edit templates**—Template expansion in Multi-Edit is one of the most important timesaving features around. Template expansion allows you to easily and consistently write repetitive tasks in your code that would otherwise bog you down with extra keystrokes. For example, when programming in C, one can type “i” and press the space bar and receive an if construct complete with curly braces and parenthesis. These templates are completely editable and are not limited specifically to code items. The



flexibility of the template system in Multi-Edit allows you to set up a template for virtually anything you can think of—everything from a comment header to an entire form letter is feasibly possible.

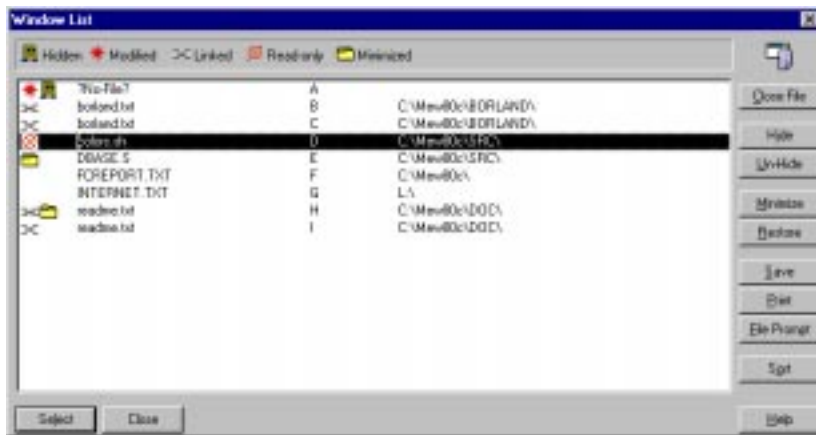


- ▼ **Language setup**—A *language* in Multi-Edit is a set of keywords, symbols, and comment characters that define a particular programming language. Multi-Edit’s language setup also supports items specific to each language (for example, how to expand curly braces for if constructs in C-style languages). **Tools | Customize | Languages** contains fields that allow you to enter keywords and other syntax highlighting options, along with a **Set Properties** button that controls language specific items.
- ▼ **Ruler**—The ruler is a useful tool for lining up your source code and quickly moving your cursor to specific columns in your files. When you activate the ruler, it will appear along the top of the current file.
- ▼ **Customize windowing**—Use this option to adjust the manner in which windows and tabs are displayed in Multi-Edit.

## WINDOW LIST BOX



The Multi-Edit Window List Box displays information about open windows, including letter names of the windows, the names of the files loaded into the windows, and the path of each loaded file.



You may also perform several window operations:

- ▼ Open a new window with **File Prompt** (causes the Open File dialog box to appear).
- ▼ Delete an Editing Window (press the <Delete> key on a selection).

- ▼ **Save** the file in a window.
- ▼ **Hide** a window to skip over it when moving sequentially from window to window and cause it not to be visible in the Next/Previous List.
- ▼ **Minimize or Restore** a window.

Each operation affects only the currently highlighted files in the Window List. The **Select** button will select the highlighted files for editing.

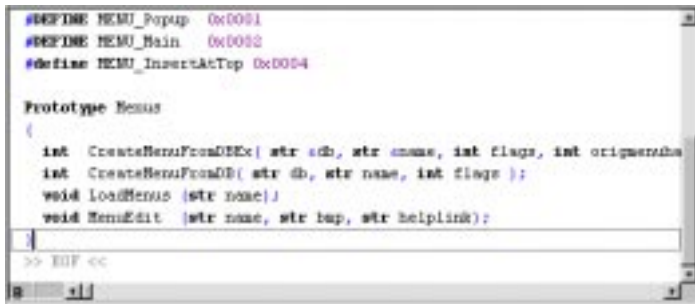
---

*NOTE: This List Box, like all List Boxes, has an incremental search feature. By typing the first few characters of a file name or list item, you can highlight that file name or list item in the List Box. Also, you can swap the position of the file name for the window letter, making the incremental search key off of the window letter. Search the Help file for Startup Macro for more details.*

---

## **EDITING WINDOWS**

Below the Tab Bar, the client area displays windows for each of the files that are being edited. The letter in the lower left corner identifies the window in the Window List, which can be displayed by clicking on this letter. Click in the area to the right of the window letter to display current file information for the active window (also available from the default **File | Information**). Use the horizontal and vertical scroll bars to navigate the contents of the window.



```
#DEFINE MENU_Popup 0x0001
#DEFINE MENU_Main 0x0002
#define MENU_InsertAtTop 0x0004

Prototype Menuus
(
int CreateMenuFromDEE( str sCb, str sName, int flags, int origname)
int CreateMenuFromDE( str sCb, str name, int flags );
void LoadMenu (str name);
void MenuEdit (str name, str lcp, str helpLink);
)
>> EOF <<
```

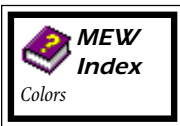
### **Context Menu**

Click the right mouse button (or press the <Esc> key) within an Editing Window to pop up the **Context Menu**. The **Context Menu** is fully configurable under **Tools | Customize | Menus**, including the addition of language-specific entries. By default, the **Context Menu** contains the following choices taken from various other menus.



- ▼ **Copy**—This selection copies the current marked block to the Multi-Edit buffer.
- ▼ **Cut**—This choice cuts the currently marked block and places it in the Multi-Edit buffer.
- ▼ **Paste**—This choice pastes the contents of the Multi-Edit buffer to the current cursor position.
- ▼ **Find tag under cursor**—This Multi-Tags operation searches for a tag with the same name as the text under the cursor.
- ▼ **Open file under cursor**—If the text cursor is sitting on a filename, you can open that file with this command. This is especially useful for opening up files listed in #INCLUDE statements.
- ▼ **Extension specific help**—Multi-Edit has the ability to access Windows help on an extension-specific basis. For example, all files with a .C extension can have Windows help files associated with it. The Extension Specific Help selection will search for a help topic with the same name as the text under the cursor. If a matching topic is not found, the Windows Help Search Dialog will remain up, showing topics that most closely match the search string. If an extension does not have a Default Help File specified, this feature will search for a matching topic in the Multi-Edit help file (MEW32.HLP).
- ▼ **Customize this files settings**—Choose from Extension setup, Edit templates, or Languages setup.
- ▼ **Edit this menu**—This selection allows you to add, remove, or modify entries in the right mouse button menu.

### **Customizing Colors**



Multi-Edit allows you to choose the colors within Multi-Edit Editing Windows. Color templates can be created for use with specific filename extensions (**Tools | Customize | Filename Extensions | Colors**).

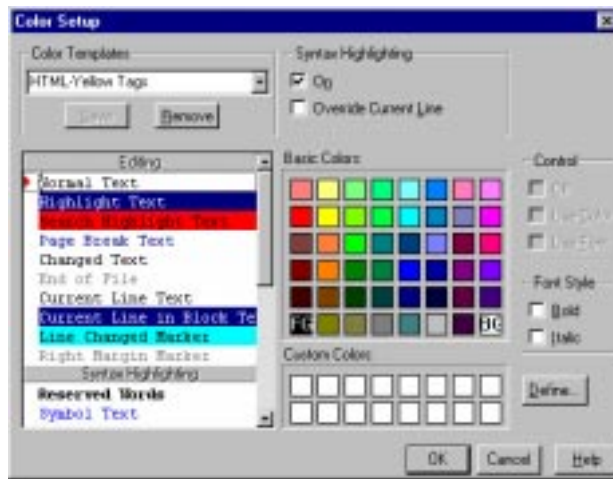
---

*NOTE: Color changes to the title and menu bars and the workspace background are set from the standard Windows color setup in the Windows Control Panel | Display | Appearance.*

---

To change the foreground and background color of the Editing Windows:

1. Select **Tools | Customize | Colors** to display the Colors Setup dialog box.



The colors dialog box also contains many other fields, which allow you to further customize your color setup.

**Color Templates**—This area allows you to define several color templates which you can select and use at any time, depending on your editing needs. You can **Save** your current color template, or **Remove** a color template. The drop-down list is provided to help you find your color templates.

**Example Text Box**—The list box below the color templates field contains examples of the different types of text that exist within Multi-Edit editing windows and how they currently look on your monitor. The top section contains Editing selections that will always be colored when in Multi-Edit. The second section represents **Syntax Highlighting** selections that will only be colored when the **Syntax Highlighting | On** check box is marked. Otherwise they will appear as normal text. The final section shows the colors that will be used during Multi-Edit's interactive file compare.

**Syntax Highlighting—On:** When checked, Multi-Edit's context sensitive syntax highlighting is enabled. If unchecked, keywords, strings, comments, etc. will appear as normal text. **Override Current Line:** When checked, syntax highlighting will override the current line color, if one exists. Otherwise, the current line color will take precedence over syntax highlighting.

**Basic Colors**—This section shows all the default foreground and background colors to choose from in Multi-Edit.

**Custom Colors**—Any custom colors that you define appear here. You can choose these colors exactly the same way as a basic color. You must define colors by pressing the **Define** button for any colors to appear here.

**Control**—Multi-Edit allows you to set some special options when defining your window colors. **Use Color:** Allows you to instruct a Syntax Highlighted item to use its currently defined color or use the normal text color instead.

**Use Font:** Allows you to use the Font Style option button field to set a bold or italic font for this Syntax Highlighted item. **Off:** Allows you to set a standard Editing item to off (disabled). Note this button is not accessible for syntax highlighting options.

**Font Style**—If the **Use Font** check box has been checked, there are additional syntax highlighting options. Bold and Italic.

**Define**—Press this button to display the Custom Color dialog box in which you can create a color and add it to the Custom Colors palette. You create a color by adjusting its hue, saturation, and luminosity or by specifying its red, green and blue values.

2. Select the screen display item whose color you want to change. FG (for foreground) and BG (for background) will appear in the Color Chart corresponding to that item's current colors.
3. The Color Chart contains numerous background and foreground color combinations. Click the left mouse button to change the foreground color to the color currently under the mouse cursor. Click the right mouse button to change the background color.

## RESULTS WINDOW



The Results Window is a tabbed pane at the bottom of the Multi-Edit 8 screen that holds tabs for Bookmarks, FTP Results, Compiler Results, Collapse Dialog, Search Results, the Project Manager and Task List.



The Results Window displays when first used. On the Status Bar, the small window icon in the lower right corner can be clicked to toggle the display of the Results Window. When displayed, the Results Window can also be minimized by clicking the minimize icon in the upper right corner of the Results Window. Use

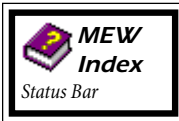
the Results Window maximize button to cause it to fill half the client area of the Multi-Edit window. When the Results Window has keyboard focus, use the <Esc> key to hide it and return to editing.

Multiple tabs can be displayed at one time and brought to the front by clicking on a tab. Close a tab by right-clicking on the tab and selecting **Close tab**. Resize the window by pausing the mouse over its top edge until a double arrow cursor appears. Click and drag the edge of the Results Window to the desired size.

### Right Click Options

Close a tab on the Results Window by right-clicking on the tab and selecting **Close tab**. In addition to this option, each tab may have options specific to its function.

## STATUS BAR



The **Status Bar** is displayed at the bottom of the Multi-Edit 8 screen.

From right to left, the Status Bar displays the following:

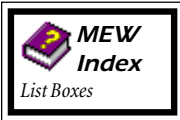


- ▼ **Status Line**—Displays messages as necessary.
- ▼ **Cursor Position**—Displays the current line and column position of the cursor in the current Editing Window. Click in this area to display the Go To Line dialog box.
- ▼ **Ins/Ovr**—Displays the Insert/Overwrite mode. Pressing this button (or the <Ins> key) toggles between Insert mode (characters typed are inserted at the current cursor position) and Overwrite mode (typing replaces characters at the current cursor position).
- ▼ **Caps**—Displays the Caps Lock status. Pressing this button (or the <Caps Lock> key) toggles the keyboard Caps Lock status.
- ▼ **Num**—Displays the Num Lock status. Pressing this button (or the <Num Lock> key) toggles the keyboard Num Lock status.
- ▼ **Rec**—Will display in red text when a macro is being recorded. Select this option to record keystrokes; when completed, select **Rec** again to stop and save the macro.
- ▼ **Results Window Status**—Pressing this button will toggle the Results Window display (minimize or restore).

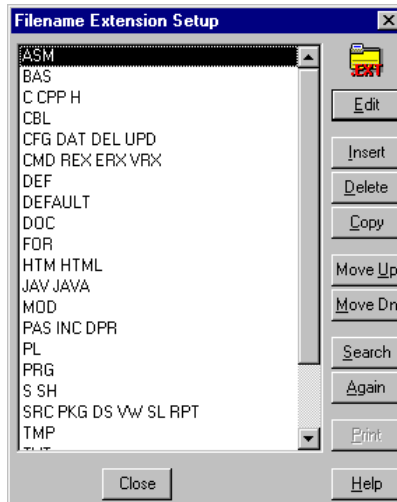
## Right Click Options

The area that displays the cursor position can be right-clicked to display the Go To Line dialog box.

## LIST BOXES



Several dialog boxes in Multi-Edit 8 appear as standard Multi-Edit List Boxes, such as the one below. Select an item from the list and choose from the buttons on the right.



# Getting Started Using Multi-Edit

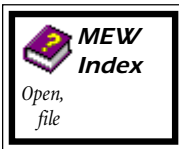
---

This section will introduce you to several Key Concepts with which you will want to be familiar in your use of Multi-Edit 8.

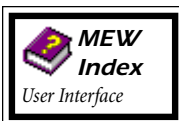
## OPENING FILES

---

### Methods



Multi-Edit offers several ways of opening a file from the default **File Menu** (Open, New, Load) and from **Window | List | File Prompt**. Selecting one of these options will display a File dialog, such as the Explorer-style dialog shown below. The Explorer-style dialog offers an interface similar to the Windows 95 and Windows NT dialogs with which you are familiar.

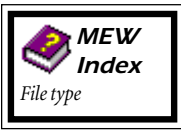


You may opt to use the Multi-Edit Classic Style dialog (select **Tools | Customize | User Interface** to configure), although we are phasing out this style. We encourage everyone to switch to the Windows 95/NT 4.0 Explorer-style dialog, as it has a lot of functionality that we will never be able to put into the Classic file dialog. See the README.TXT file for more information.

Files may be quickly selected for editing by choosing from the list of most recently opened files at the bottom of the **File Menu**. The list will contain a maximum of nine files on screens more than 480 pixels high; otherwise, a maximum of four files will be displayed.

When the cursor is on a filename in an Editing Window, select **File | Open file under cursor** to open the file in a new window.

## File Types



When opening a file, the **Type** field allows you to change the line terminator for the file. Multi-Edit uses this “file type” to determine both where to break lines when loading a file and which characters to insert when the <Enter> key is pressed during normal editing. DOS files use a carriage return and a line feed to mark the end of the line. UNIX, uses only line feeds to mark the end of the line. Binary files have no line terminators—they are continuous streams of data; however, for easier reading, binary files use a Binary Record Length that can be specified on the fly or for a particular extension.

Each extension can have its own default file type. By changing the **Type** field when opening a file, it is possible to override the default file type. It is also possible to tell Multi-Edit to automatically detect an extension’s file type when loading it. The default file type is set in **Tools | Customize | Filename Extensions**.

## Saving Files



Multi-Edit offers several methods to save files. **File | Save** will cause the current file to be saved to disk under the existing filename. If the file has not been previously saved, you may select a filename and other options. Use **File | Save As** to save a file for the first time, or to change the filename and other options. **File | Save All** allows you to save all modified files in open windows. Use **File | Close** to close the current window and save modified files. Select **File | Close All** to close all Editing Windows and save modified files. You will also be prompted to save modified files when exiting Multi-Edit.

Options for Autosave and Backups can be configured in the **Tools | Customize | Files** dialog.



## VIEWING FILES

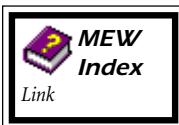
Normally, Multi-Edit uses the standard MDI style windowing arrangement (each window contains one file with that file staying in the same window). When using single window mode (configure in **Tools | Customize | Windowing**), a window is a view in which each file is cycled through, (this will be familiar to Brief users).

### Organizing Windows



The **Window Menu** provides a variety of commands for navigating between, sizing, and manipulating windows. Several of these options are also available from the Multi-Edit Window List. **Hide** conceals the current window. After that, **Next** and **Previous** will skip over it. To UnHide windows, select **Window | List**, highlight the desired window and press the **UnHide** button, or select it from the **Window | “Quick Pick” Window List**. **Window | Zoom** allows you to instantly maximize a window within the Multi-Edit environment or return it to its previous size. Mouse users can achieve the same results by clicking the left button on the Minimize/Maximize buttons, or double-clicking on the window’s title bar.

**Window | Cascade** provides ways to group your windows in a cascade arrangement (like tabbed index cards). **Window | Tile vertical** allows you to arrange your non-minimized windows into a vertically tiled arrangement. Use **Window | Tile horizontal** to arrange your non-minimized windows into a horizontally tiled arrangement.



Select **Window | Split** to divide your current window in half, having two adjacent windows occupying the screen area previously occupied by the original window. You can use **Split** to view two or more files at the same time, or view two or more parts of the same file simultaneously. The latter is called *linking*.

Windows created with **Split** differ from full screen windows in one important aspect. All “children” of the original window share what we refer to as a *virtual screen*. All windows associated with the same virtual screen are updated simultaneously—if one is redrawn, they all will be. For this reason, they are tiled and cannot overlap.

When **Split** is invoked, a dialog box with four arrows appears. Each arrow points in a different direction. Your arrow choice determines where the next window appears. After choosing where you want a split window to appear, the Link Window dialog box appears (a standard Multi-Edit Window List). Linking is optional. Select a window from the list to which you would like to link the first window, or press <Esc> to cancel linking. If you decide not to link, the original window’s file will be loaded into the split one.

## ***Navigating Windows***

Windows in Multi-Edit are identified by a window letter, assigned in alphabetical order as the window is opened. **Window | Next** moves you to the next window in alphabetical order. **Window | Previous** moves you to the previous window in reverse alphabetical order. Hidden, minimized and compiler error windows are skipped over during navigation.

**Window | Link** allows you to have two or more windows containing the same file. You may edit in any one of the linked windows, and the changes will be reflected in the other(s). This is very handy if you need to edit a file in two or more places. **Link** will display a standard Multi-Edit Window List from which you can choose the file to which you wish to link.

---

*NOTE: By default, files loaded multiple times in Multi-Edit are not linked.*

---

## ***Display Options***

Text can be displayed in Hex Mode and with Line Numbering.

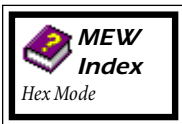
### ***Hex Mode***

Select **Hex mode** from the **Text Menu**. From the **Tools | Customize | Editing**, you can configure Multi-Edit to edit binary files in hex mode by default. Selecting hex mode results in a side-by-side split: the left side is in hex, the right side is in ASCII. When editing in the left side, characters may only be entered in hex, with the overwrite mode always on.

Hex mode is simply a different view of the current file. It does not assume that the file is binary, nor does it change the “file type”. Thus, template expansion, smart indent and other features work (if they are configured for that file) while you are editing a file in hex mode. If you wish to view line terminators for a file then you will need to load it as a binary file.

### ***Line Numbering***

**Text | Line Numbers** displays line numbers along the left margin of the Editing Window.



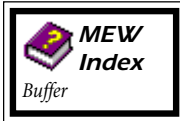
---

## EDITING TEXT

---

These concepts and functions are important to understand when using Multi-Edit.

### **The Multi-Edit buffer**



The Multi-Edit buffer can be regarded as a temporary storage area to place a block or blocks of text, which can then be retrieved at a later time. The Multi-Edit buffer is similar to the Windows clipboard, except that the Multi-Edit buffer holds all cut and copy operations for the entire session. When a block of text is cut or copied to the Multi-Edit buffer, it is also copied into the Windows clipboard. Select **Edit | Show buffer** to view the contents of the Multi-Edit buffer.

### **Editing Operations**



The **Edit Menu** offers convenient tools for cutting and pasting text to and from the clipboard, as well as **Undo**, **Redo**, and **Repeat**. Multi-Edit offers increased functionality over these standard Windows commands.

#### ***Cut, Copy, Append, Cut and Append, Paste***

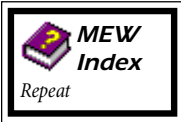
**Edit | Cut** deletes the marked block from the current window and moves it to the buffer. **Edit | Copy** copies the marked block to the buffer. **Edit | Append** copies a marked block and adds it to the end of text already in the buffer. **Edit | Cut and Append** deletes the marked block from the window and adds it to the end of any text already in the buffer. **Edit | Paste** copies text from the buffer and places it at your cursor position.

#### ***Undo and Redo***

**Edit | Undo** allows you to cancel any text change you made. For example, you can restore a character you deleted, reverse the effects of a Search And Replace operation, or return a marked block to its previous position after its been moved. Undo may be selected repeatedly. It has the ability to cancel up to 65,000 changes (should you choose to configure that many changes in **Tools | Customize | Editing**).

**Edit | Redo** reverses the effects of the **Undo** command. You can redo as many changes as were “undone” with the following rule:

After a change is made to a file following a redo, previous undo operations can not be reversed. For example, suppose you used Undo 10 times and then used Redo 3 times. If you then make a change to the file, you will not be able to redo the remaining 7 Undo steps.



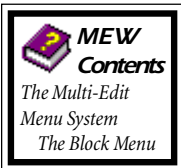
### ***Repeating Keystrokes***

**Edit | Repeat** allows you to repeat a single keystroke (whether assigned to a command or not) many times in succession. Here is how the **Repeat** command works:

- ▼ Select **Edit | Repeat** or press <Ctrl+R>. A message appears on the Status Bar prompting you to enter a number.
- ▼ Type the number of times you want the keystroke to repeat.
- ▼ Type the desired keystroke. The keystroke will be performed the number of times you have indicated.

If you type 0 (zero) for the number of times you'd like the keystroke to repeat, Multi-Edit will repeat that keystroke until the cursor reaches the End of File marker. You should ensure that the keystroke you enter will cause the cursor to reach the End of File marker. If, by mistake, the repeat command becomes stuck in an infinite loop (will never reach the end of the file), use <Ctrl+Break> to terminate the macro.

### ***Block Operations***



The **Block Menu** provides commands for marking blocks of text, and for moving and copying text you have marked. Several block operations can be performed with the improved functionality of the mouse in Multi-Edit. In addition to those reviewed below, block operations include **Delete**, **Window Copy**, and **Window Move**.

The following are accessible from **Block | Block Operations**.

#### ***Clear Block***

Performs simple clearing of a marked column of text, converting all text contained within the marked block to blank lines or blank characters.

#### ***Fill Block***

Fills a columnar block with a character or a series of characters by prompting for the characters to use, then duplicating the characters downward to fit the block. If you select "Fill entire width of marked block" then the block will be filled, repeating the characters, until the entire width and height of the block is full.

#### ***Fill Block Down***

FillDown is used to make multiple copies of the top line contained within a block. Start by marking the area of text you want to work with (at least two lines) using a Line or Column block. By invoking FillDown, the top line of text contained within the marked block is copied downward to fill the block.

**Fill Block Up**

FillUp is the opposite of FillDown with the exception that the bottom line of text contained within the marked block is copied upwards to fit the block.

**Fill Paragraph**

FillParagraph will duplicate the currently marked block or current unmarked line to the next available line, paragraph, or blank line.

**Fill Series Down**

SeriesFill will enumerate a block of text. It works by obtaining values from the first occurrence of a whole number from the first and second lines within the block. If the first number is 20 then numbering continues upward from that number applying an increment based on the difference between the first and second numbers. For example, when the first line begins with 20 and the next is 21 then subsequent lines are numbered 22, 23, 24, and so on. If the second line begins with 22, however, subsequent lines are numbered 24, 26, 28, and so on.

---

*NOTE: If "/M=1" is omitted from the command line then a dialog will show allowing custom increment and start values, hex conversion, etc.*

---

**Fill Smart**

SmartFill evaluates the selected area of text and determines what sensible fill concept to apply.

**Case A:**

- ▼ If no block is marked or only one line style block marked.
- ▼ If the above conditions are true then FillParagraph is performed on the current line.

**Case B:**

- ▼ If at least two lines are marked.
- ▼ There are numbers contained within in the marked block.
- ▼ The numbers are not yet enumerated.
- ▼ If the above conditions are true then a FillSeries will automatically be performed with one exception. If the increment value for SeriesFill is 0. Then SmartFill will apply 1 as the increment value.
- ▼ Exception: If at least two lines are marked, and there are numbers contained within in the marked block. And the numbers are not yet enumerated, and all the lines selected below the first two lines are blank. Then SeriesFill will

FillDown the remaining lines while still applying a SeriesFill using the pre-determined increment value.

***Case C:***

- ▼ If at least two lines are marked.
- ▼ There are numbers contained within in the marked block that are already enumerated, or there are no numbers contained within the marked block.
- ▼ The cursor is sitting on the last line of the marked block.
- ▼ If the above conditions are true then a FillDown will be performed.

***Case D:***

- ▼ If at least two lines are marked.
- ▼ There are numbers contained within in the marked block that are already enumerated, or there are no numbers contained within the marked block.
- ▼ The cursor is sitting on the first line of the marked block.
- ▼ If the above conditions are true then a FillUp will be performed.

***Case E:***

- ▼ A columnar block is marked.
- ▼ Contains no numbers.
- ▼ Is blank or was already filled using FillDown or FillUp.
- ▼ If the above conditions are true then FillBlock will be invoked.

***ConvertBlock***

The ConvertBlock macro will prompt to convert a stream block to a line or column style block. All the Fill features listed above use this macro.

***Persistent Blocks***

Select **Block | Persistent Blocks** to toggle the Persistent Blocks mode on and off.

Most operations in the **Block Menu** are oriented toward Persistent Blocks, meaning that once a block is marked, the block will stay marked until you delete it or turn the marking off. This allows you to mark a block, move your cursor to another place in the file, and copy or move the block to the new location.

The Cut and Paste clipboard operations explained in the Editing Text section are oriented toward Non-Persistent Blocks. In this mode, the block marking will turn off whenever the cursor is moved. Also, the following keystrokes will take effect (configurable in **Tools | Customize | Blocks**):

- ▼ <Delete> or <Backspace>—Will delete the marked block.
- ▼ <Tab>—Will indent the marked block.
- ▼ <Shift+Tab>—Will unindent the marked block.

### Marking Blocks

Double clicking with the left button on a word in an Editing Window will cause that word to be marked as a columnar block. Triple clicking will cause the whole line to be marked. You may also use **Block | Mark Lines of Text** to set starting and end points for marking lines of text.

To mark a stream block, position the cursor at the desired starting point. While holding down the left mouse button, move the mouse cursor to the desired ending point and release the left mouse button. You may also use **Block | Mark Stream of Text** to set starting and end points for marking streams of text.

Another way to mark a stream is to place the text cursor to the desired starting position, then click on the desired ending point and while pressing the <Shift> key. A block will be marked from the starting position to the ending point. Extend the block marking by pressing the <Shift> key while clicking at the desired ending point for the block.

While marking a block with the mouse, you can hit the right mouse button (while still holding down the left button), and the block mode will change. Repeatedly clicking the right mouse button will cycle through the line, stream, and column marking modes. You may also use **Block | Mark Column of Text** to set starting and end points for marking columns of text. Whenever you use the mouse to mark a block, a cursor will be displayed next to the mouse cursor indicating the type of block being marked.

---

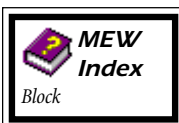
*NOTE: By default, the mouse marks streams of text.*

---

### Moving and Copying Blocks

To move a block with the mouse, click and hold the left mouse button on any position inside a marked block, then drag the mouse cursor to the position to which you want to move the block. You will notice a small icon displayed next to the mouse cursor, indicating that a block move operation is occurring. When you release the mouse button, the block will be moved to the new position.

To copy a marked block to another location without removing it from the first location, press the <Alt> key while you click and hold the left mouse button on any position inside a marked block, then drag the mouse cursor to the desired position and release the mouse button and <Alt> key.

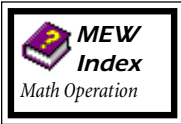


### ***Save Block to Disk***

**Block | Save Block to Disk** allows you to save the marked block to a file all its own. Specify the file in which to save the block in the Save Block As dialog.

### ***Math Operations on Blocks***

**Block | Math Operation** is used to perform addition, subtraction, multiplication or division on a marked column of numbers. If you have Persistent blocks selected, the result of the Math Operation will be placed at the current cursor location. If you have non-persistent blocks selected, the result of the Math Operation will be placed in the Multi-Edit buffer. You can then paste the contents of the Multi-Edit buffer wherever you wish.



---

## ***FORMATTING TEXT***

---

Multi-Edit excels in its formatting functionality, including the ability to indent blocks of text, change case, fill white space with tabs or spaces, automatically format comments, center lines and justify paragraphs.



### ***Indenting Blocks***

**Block | Indent** moves all the lines in a marked block one tab stop to the right. **Block | Undent** moves lines in a marked block one tab stop to the left.

### ***Change Case***

Select **Text | Change Case** to change the case of text on a line, in a word, or within a marked block: Lowercase, UPPERCASE, Proper Case, Toggle case.

### ***Tabs vs. Spaces***

There are two methods for placing white space in a file: all hard spaces or tabs, based on tab stops or format line and spaces to fill (configurable in **Tools | Customize | Editing**). If you select tabs, whenever a <Tab> key is pressed or white space added, an actual hard tab will be added, saving space in the file. This can lead to formatting problems when the file is edited by someone who doesn't have the same tab settings. Using spaces to fill in white space ensures exact spacing, but increases file size.

---

*NOTE: The default option can be overridden by an extension-specific setting (Tools | Customize | Filename extensions). When using tabs or spaces, the size of a tab stop is controlled by a hard number of columns per tab stop or a format line. Again, these can be defaulted and overridden by each extension-specific setting.*

---

## Commenting

The Comment/Uncomment feature (available from the **Tools Menu**) works for any language for which you have set up comment characters. It works best if you have set up both open/close comments and end of line comments, but it will work just as well with only one comment style defined. To set up comment characters, use **Tools | Customize | Languages**.

The comment feature has several functions built into one command. Which function gets performed depends on how you use the features. There are four methods of using the comment/uncomment command. Each method has two different behaviors depending on the conditions when using the feature.

- ▼ Method 1—No block marked: If you invoke the comment/uncomment command when no block is marked, Multi-Edit will comment out the line that the cursor is sitting on. If that line has already been commented out, it will uncomment that line.
- ▼ Method 2—Line block marked: When a line block is marked and you invoke the comment/uncomment feature, Multi-Edit will comment out the marked lines much the way it does with Method 1. Again, if the marked lines are already commented, it will uncomment them.

---

*NOTE: Multi-Edit cannot "reverse" comments. Thus, you cannot comment one line and uncomment another at the same time.*

---

- ▼ Method 3—Stream block marked: When you mark a stream block and then invoke the comment/uncomment command, Multi-Edit will comment your code from the beginning of the stream block to the end, and vice-versa if the marked code is already commented.

---

*NOTE: If you have only defined end-of-line comments for the language you are using (or begin/end comments do not exist), then you may not be able to end your comment at the point specified in your stream block (i.e., your last line will be completely commented out, since only end of line comments exist).*

---

- ▼ Method 4—Single column of text marked: If you mark a single column of text and then invoke the comment/uncomment feature, your code will be commented starting at the column marked and extending to the end of each line.

## Formatting Lines



**Text | Center Line** lets you center the line at the current cursor position between the first column and the right margin. The Center Line command does not dynamically center text as you type it. To use **Center Line** correctly, first type the text you want to center, then center the text. If you add to that text later, you may want to re-center it to correct for unbalanced space on either side of the centered text.

---

*NOTE: To avoid surprises, it is a good idea to know in advance where you have defined your right margin.*

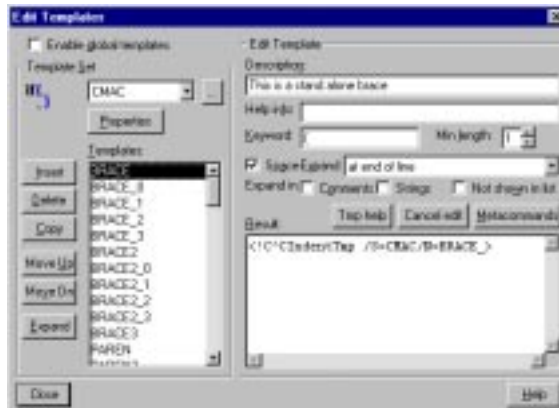
---

**Text | Justify Paragraph** inserts spaces in each line of a paragraph until their ends are flush to the right margin. This command will start justifying from the cursor position to the end of the paragraph.

## TEMPLATES



Template editing allows you to create customized sets of templates on the fly for programming languages, documentation, etc. The **Tools | Edit Templates** dialog box is modeless, which means you can leave the box up and edit simultaneously.



The template data for each template set is stored in a special file with an extension of .TPT. Others can be created easily. Template sets are associated with a file extension in much the same way as a language type. You can set the template set for each extension by selecting **Tools | Customize | Filename Extensions**.

Global templates can be used in the same way as regular templates (i.e., expanded via a keyword and/or via a template expansion metacommand from any other template set). If enabled via this checkbox, global templates are available to any Editing Window regardless of file extension. Templates provide a way to reduce the amount of typing needed when entering large amounts of repetitive code. Some good candidates for templates are flow control statements, documentation headers for routines and API functions. For example, with a routine header a template could be created that enters the basic outline of the header with template field markers placed where the routine-specific information would be entered and, with a few keystrokes, would be fully expanded with the cursor positioned on the first field marker.

### **Enabling Templates**

Templates are organized in what are called *template sets*. Each template set is stored in a file with an extension of .TPT. There are three classes of templates sets: language templates, global templates and other templates.

Every language that Multi-Edit supports comes with a template set that contains templates specific for that language and is named after the language (i.e., C.TPT, PASCAL.TPT and JAVA.TPT). A language template set is specified in the **Template** field of the Extension Setup dialog for each extension (**Tools | Customize | Filename extensions | Edit**). Only one language template can be assigned to an extension, although other templates and global templates can still be used.

All global templates are stored in the file named GLOBAL.TPT and, when enabled, are available to any Editing Window regardless of the file extension. These templates would be templates that contain text that you would enter into any type of file such as your name or company name, etc. To use global templates you must enable them by checking **Enable global templates** in the **Tools | Edit Templates** dialog and **Auto-template expansion** in the Extension Setup dialog (**Tools | Customize | Filename extensions | Edit**) for each file extension for which global templates are to be used. When enabled, global templates will be searched for after the language template set is scanned. Thus, a template in a language set by the same name as a global template will be expanded instead of the global template.

Other templates are templates that may not be specific to any language such as templates for the Win32 APIs or templates that are specific to a vendor's language product such as Borland C++ runtime libraries. Any number of these template sets may be used and are set up by entering their names, separated by semicolons, in the **Addon templates** field of any Extension Setup dialog.

## ***Invoking Template Expansion***

Template expansion can be invoked by selecting **Tools | Build Templates** or by enabling **Auto-template expansion** in the Extension Setup dialog. This causes the space key to trigger template expansion.

Template expansion can be performed at any time while editing text. When invoked, the template system checks to see if there are characters before the cursor that match a defined template. If a unique matching template is found, it is automatically expanded; otherwise, a list of possible templates is shown. From this list, select a template and press **Expand** to cause the highlighted template to be expanded at the cursor location.

The toolbar button method of template expansion can be demonstrated by using the HTML templates. When one of the HTML template toolbar buttons is selected, the template system expands the specific template for that button. These templates are special in that they have been designed to remove a block of marked text if it exists, insert the template text and paste the contents of the removed block into the correct place in the text. These templates can only be expanded from the toolbar button or selecting them from the Build Template list.

One feature built into templates that can be a great time saver is *field strings* or *field markers*. A field marker is text in the template between back tick characters. The purpose of these markers is twofold. First, they document what is expected at its location when entered in templates with many fields. Most importantly, the field markers provide a quick way to navigate through a template as it is being filled in.

In the default keymap, the <Ctrl+I> and <Ctrl+U> keys are assigned to functions that move the cursor to the next and previous template field marker. Using these keys allows you to move the cursor quickly to the next field marker and have the field marker block marked and deleted when text is typed in.

Of course, not everyone likes the field markers being entered, so templates that do have field markers can be set to have the field markers stripped when the template is expanded. This is accomplished in the **Tools | Edit Templates** dialog by selecting **Properties** and then checking **Strip field strings**.

## ***Editing Templates***



All templates are user-configurable and new ones can be easily added. To change or add templates use **Tools | Edit Templates** to bring up the Edit Templates dialog where you can edit, add, delete and create new template sets.

## SEARCH OPERATIONS



The **Search Menu** provides commands that allow you to search for text, search and replace text, save and retrieve cursor positions, and move to a specific line number. All of the search dialogs have been merged into a single, tabbed dialog. In addition, for consistency with other Windows applications, the word “Search” has been replaced with “Find” in many instances.

Being able to easily find and replace text, save a position in a file and quickly return to that position is essential in a good programmer’s editor. Multi-Edit provides many ways to accomplish all of these through the **Search Menu**.

Besides just finding literal strings, Multi-Edit allows patterns, called *regular expressions*, to be entered and found. Of special note are regular expression aliases discussed later in the section.

You, as a programmer, are constantly searching for text. You may need to know how a variable or function is defined and how it is used earlier in a file. Multi-Edit makes it easy to find what you are looking for, presenting it in different ways, to give you more time to analyze and improve your code.

### Global Expression Highlight



**Search | Global Expression Highlight** allows you to specify a search string or expression and see all occurrences of the found text highlighted in all of your files.



This is dynamic; you may edit the files as usual without disturbing the highlighting. This function allows paging through a file to quickly see how a string or variable is used.

*NOTE: There is a limitation to Global Expression Highlight. Search strings can not go across line boundaries.*

## Find Word Under Cursor (*SearchWord*)



A quick way to find the next or previous occurrence of the word under the cursor can be done by using the <Ctrl+Alt+Up> or <Ctrl+Alt+Dn> keys available in most keymaps or by using <Shift> or <Ctrl> with the right mouse button. If the word under the cursor is not already in a block mark then hitting one of these keys will block mark the word and will move to the next or previous matching word on the next keystroke. A phrase can also be searched for in this manner, but it will need to be block marked before hitting the search keys.

## Incremental Search



Multi-Edit's powerful incremental search feature allows you to perform simple searches with ease. Place the cursor at the position from which you wish to begin searching, select **Search | Incremental Search**, and begin typing the string of characters for which you would like to search. Multi-Edit will search for the string as you type it in, highlighting the next occurrence of the search string in the file. Thus, you may not need to input the whole string to find the text for which you are searching. Once a string is found, the next or previous occurrence of that found string is located by using the <Alt+N> or <Alt+P> keys.

## The Find Dialog



The Find dialog is used to find strings in the current file or any file currently loaded into Multi-Edit. Use **Search | Find** to bring up the Search tabbed dialog with the Find tab enabled.



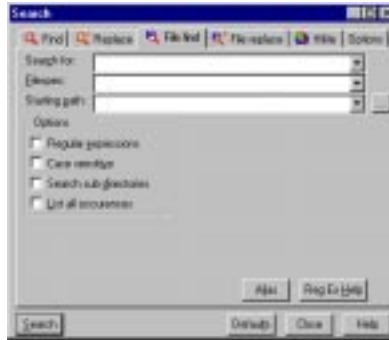
Enter the text you are searching for, set the options and then hit the **Search** button to start the search for a single match. To get a list of all matching strings, use the **All** button instead of the **Search** button; the Find tab will appear in the Results Window with a list of all matching lines in the current file (or all windows

if **Search all windows** is checked). If you wish to search all the files specified in the current project, be sure to select **Search project** before selecting the **Search** or **All** buttons; Multi-Edit will do a **File Find** on all files in the project.

### **File Find**



The File Find dialog is used to find files matching a wildcard, or multiple files containing a specified string. Using **Search | File Find** will bring up the tabbed dialog with the File find tab enabled.



To find files matching a wildcard, fill in the **Filespec** and **Starting path** fields, leaving the **Search for** field blank. Set the desired options and select the **Search** button. This will cause Multi-Edit to search for and list only the filenames matching the wildcard specification. The results will be shown in the FS List tab of the Results Window. To find files that contain a specific string, do everything the same as you would for finding a filename only, but also fill in the **Search for** field with the string to be found. By selecting **List all occurrences**, all matching lines will be shown in the Results Window.

### **Replacing Text**

Replacing text is accomplished through one of two dialogs. Which dialog you select depends upon whether you want to replace text in a file loaded into Multi-Edit or ones that are not loaded.

#### **Replace**

**Search | Replace** is used when replacements are to be made in the current file or any file loaded into Multi-Edit.

<p><b>MEW Contents</b>          The Multi-Edit Menu System          The Search Menu          Search            Replace</p>
--



Similar to the find dialog, there is an additional text field for the replacement text and functions. The **Replace** button is now active and the **All** button causes all matching strings to be replaced with the replacement string.

### File Replace

**Search | File Replace** is used when replacements are to be done in multiple files, including those that are not currently loaded into Multi-Edit.

<p><b>MEW Contents</b>          The Multi-Edit Menu System          The Search Menu          Search   File Replace</p>
--



It is similar to the File Find dialog, but after all of the matches are found and listed in the Results Window, the files are then opened and the replacements can be performed.

*NOTE: All find dialog options and search/replace strings are saved in the Session information.*

## Marking and Retrieving Position

Multi-Edit supports two types of marking positions in text, a *marker stack* and a *random access mark*. Each file can contain up to ten of each type of mark and are saved and restored across sessions.

### Marker Stack

The marker stack is useful for saving a position in a file so that you can quickly return to it after moving elsewhere in the file. The system uses the marker stack for its own use such as when doing a File Compare or matching a language construct (this should not interfere with the user dropped marks).



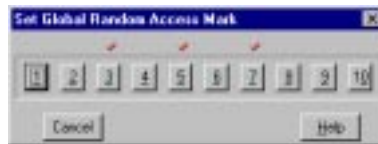
Select **Search | Push Position onto Marker Stack** to drop a mark at a position. When a marker is dropped, a little red check mark appears in the left-hand column of the line to notify the user of the dropped mark.

To return to a marked position, select **Search | Get Position from Marker Stack**. This causes the cursor to be located at the saved position and the little red check mark will be removed.

### Random Access Mark

Random access marks allows you to drop a mark and reposition to that mark in any order. This is useful for switching between multiple places in a file.

To drop a random access mark, select **Search | Set Random Access Mark**. A dialog with ten numbered buttons will be shown; select one of the numbered buttons to cause that number mark to be dropped.



When this happens, a small red number corresponding to the dropped mark will appear in the left-hand column to indicate the location of dropped mark. A little red check mark will appear above all of the numbered buttons that have active marks dropped.

To return to a saved position, select **Search | Retrieve Random Access Mark**. This will bring up a similar dialog as the one for dropping a mark. Select a numbered button to cause the cursor to be restored to the position saved for the selected mark. The little red check mark above a numbered button is used to indicate which random access marks have been dropped.

To clear any or all random access marks in a file, select **Search | Clear Random Access Mark**. This will display a dialog similar to the ones used for setting and retrieving random access marks. Now, when selecting a button, the

number mark is cleared and removed from the current file. The **Clear All** button removes all random access marks from the current file. When a random access mark is cleared, the little red number in the left-hand column is removed, as well as the little red check mark above the number button.

### Bookmarks

The Bookmark Results Window is alternate method of viewing random access marks, allowing the association of a string to each mark, making it easier to remember what each mark is marking. When **Global** is checked, it will allow saving and restoring marks across files. The Bookmark Results Window is opened by selecting **Search | Bookmark**.



---

*NOTE: Bookmarks are saved in the Session information.*

---

### Goto Line Number

**Search | Goto Line Number** displays a dialog where you can enter a line number to which to reposition the cursor.

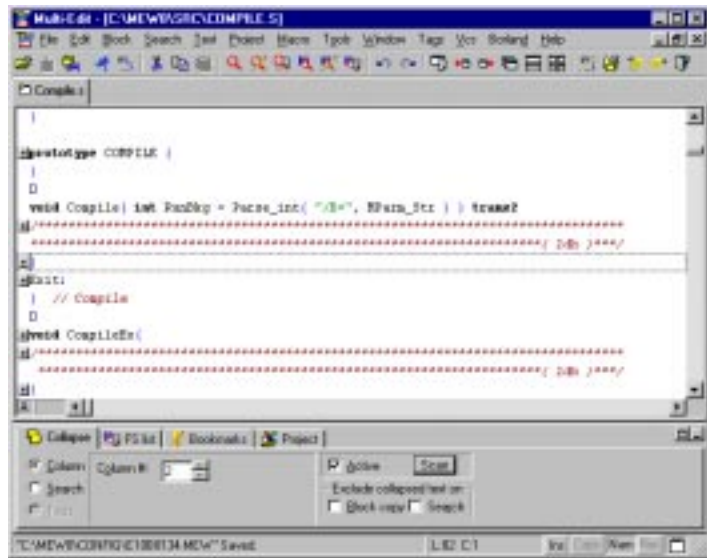
### Using Regular Expression Aliases

Search Aliases in Multi-Edit allow you to set up predefined regular expressions to use in your search strings. They can also be considered user-definable metacommands. To use an alias, you can either manually insert the alias (such as <a> for alphanumeric), or you can bring up the list of aliases, highlight one, and press the **Select** button. The real power of the Alias feature is the ability to create your own aliases.

## COLLAPSING TEXT



Select **Text** | **Collapse** to display the Collapse tab in the Results Window.



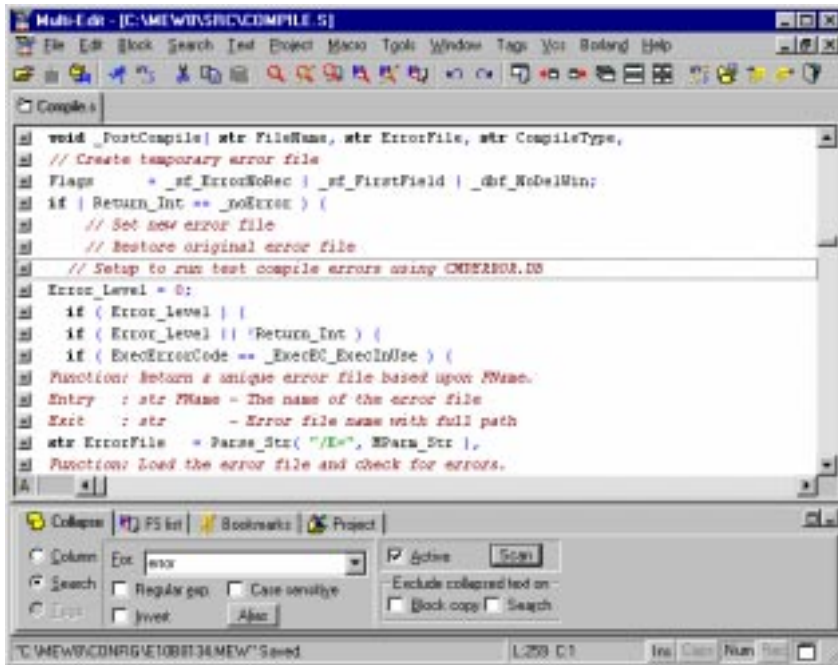
Selecting the **Column** option enables collapse mode by column number. The column on which to collapse can be selected in the Collapse Results Window. The **Scan** button takes the current collapse mode settings and collapses (or re-collapses) the current file according to those settings.

Collapse Mode allows you to view and edit a file in an outline-style format. You could, for example, choose only to view lines that start at columns less-than-or-equal to 8, or perhaps you only want to see lines that contain the word “ERROR”.

The difference between using Collapse Mode and performing a Find All operation is that Collapse is more interactive, allowing you to quickly edit the viewed lines, selectively un-hide the hidden lines, and perform block operations that either include or exclude hidden lines.

Lines that do not meet the collapse mode parameters are hidden and a plus character is shown to the right of the line where this hidden (or collapsed) text resides. To view the collapsed text, click on the plus button. The text will be expanded, and a minus button will replace the plus button. When you are ready to re-collapse your text, click on the minus button.

You can have separate collapse mode settings for each open file within Multi-Edit. Thus, you can have one file set to collapse on column 1, while another file collapses based on the keywords PROCEDURE or FUNCTION.



Select the **Search** option to enable collapse mode by search string. Additional fields pertaining to search strings appear in the dialog.

## MULTI-TAGS



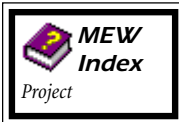
Multi-Tags is an easy-to use, hypertext-like source code browser.

- ▼ Run your source files through the Multi-Tags scanner to produce a database of functions/procedures, structures, types, etc., depending on the language being scanned.
- ▼ Once the database is created, position your cursor on any function name (or other supported language object) and select **Tags | Find tag under cursor**. Multi-Tags will then locate the source file where that tag was defined and take your cursor to the definition. If a tag was defined in more than one place or file, then **Tags | Find again** will locate the next occurrence.
- ▼ **Tags | List tags** pops up a list of the current tag database that allows you to select any tag and move immediately to its definition.

- ▼ **Tags | Browse current file** allows you to see only the tags for the file in your current window.

Currently, C/C++, CMACW, Turbo-Pascal, Java, Fortan, ASM, Modula-2, DBASE/XBASE/CLIPPER and PARADOX are directly supported. Any text file may contain tags via explicit tags.

## PROJECT MANAGEMENT



Project Management allows for a containment of the many specific settings related to a particular project. A project contains a list of files for easy access, including non-text files, directories, tools, and FTP configuration. All Project options take precedence over other options (i.e., Filename extension setups).

To set up a project, select **Project | Set**.



Select **Add** to display a File Open prompt from which you can either pick an existing project type in the name of your new project file. Multi-Edit will keep track of all your project files. Once added, **Select** the project.

*NOTE: If you need to delete the project you can just remove it from your list; however, this does not actually delete the project file. This allows for sharing of projects by using the .PRJ Project File.*

Once the project has been set it, is now possible to view all files and options by selecting **Project | View**. This will add a tab to the Results Window at the bottom of the screen.



From here you can add and remove files, open files, perform FTP transfers, sort the display and set options.

*NOTE: It is important that you configure the **Project | Options | Directories**. See the Help System for details.*

While the Project Manager keeps information on each individual project, it does not contain information on the files that were last opened or a search history. These can be done with the Session Manager. When switching Projects, you should also switch to a matching session. This can be done automatically by setting the **Tools | Customize | Projects | Synchronize Projects to Session** option. Encoded sessions must be enabled in **Tools | Customize | Sessions**.

## SESSION MANAGEMENT



A *session* is a history of what was last done in Multi-Edit. This include things such as which files are open, how the windows are arranged, history of strings used in search and/or replace operations, the location of the cursor in each file, and what project was being edited.

Sessions can be stored in several different ways, configurable in **Tools | Customize | Sessions**.



- ▼ **One global status file**—Global Session allows the status of the editor to be restored the same every time, allowing for only one status to be stored.
- ▼ **Status file in each dir**—Status file in each directory which is only useful if you are starting Multi-Edit from a different working directory each time, either by having different shortcuts or running from the command line. This stores the status based on the current working directory.
- ▼ **Encoded status files for each dir**—Encoded status files (the requirement for using the Session Manager), keeps an encoded list of status sessions. These sessions can be restored by picking from the named list in the Session

Manager or giving command line options to restore a particular status session.

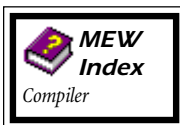
Status files are stored in the directory specified in the **Status file path** field. If this is blank, the \CONFIG directory for the user will be used. If **Status file in each dir** is used, the status files will be stored in the current working directory no matter what.

When using the encoded status files option, the Session Manager provides you with a central place where each working environment can be tracked.

Descriptive names for each environment, such as “Database Project”, store all editor environments related to that editing session. Multiple projects can each have different layouts and settings. This allows you to configure work environments for each project or session. The session list can be sorted and sessions can be protected so that environment settings will not change. For example, Session X has two files open. If it is protected and another file is opened, the next time the session is started only the two original files would be opened. If the session is not protected, all three files would be opened. Encoded status files are files stored with a .MEW extension.

Command line options are available to allow you to start a named session, the last session (which is the default), or to bring up the Session Manager on startup to allow you to pick from the list of sessions.

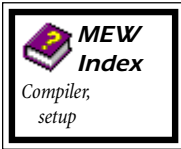
## COMPILING WITHIN MULTI-EDIT



Since programmers spend quite a bit of time in their editor, Multi-Edit includes an interface that allows you to easily run the compiler or other programs from within Multi-Edit. As with most things in Multi-Edit, it is totally user-configurable and is tied to the **Tools | Filename Extensions | Edit** setup.

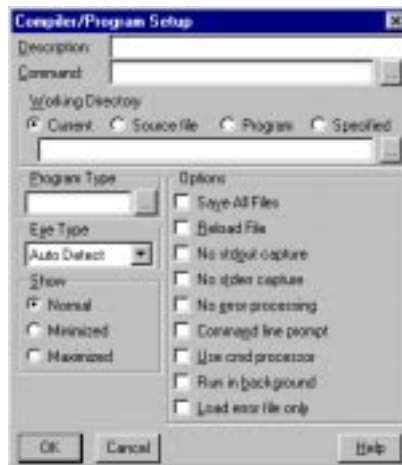


## Compiler/Program Setup



Each filename extension may have any number of compiler/program interfaces associated with it. Each compiler/program interface contains its own command line and configuration. Thus, you might have multiple compilers, linkers and debuggers set up for .C extensions. You might have a grammar analyzer setup for .DOC files.

Select **Customize** from the **Tools Menu** and click on the **Customize** tab. Press the **Filename extensions** button to display a standard Multi-Edit list box with a list of existing Filename extensions. Select the filename extension from the list and press **Edit**. Press **Compiler/Program setup** to display a list of existing Compiler/program configurations. Press **Edit** or **Insert** to display the Compiler/Program Setup dialog.



## Executing the Compiler



**Tools | Execute Compiler** allows you to use your favorite compiler to compile source code in a file without leaving Multi-Edit. The compiler you use and the manner in which Multi-Edit works with it depends on the Filename Extension Setup of the file being edited.

Multi-Edit is designed to support any compiler. You can add your own compiler support (see the Help System for details); if you have trouble, contact Technical Support by email: [tech@multiedit.com](mailto:tech@multiedit.com).

When selected, a standard Multi-Edit List Box appears containing compiler interface menu items from which you may select, add and modify compiler interfaces (see the previous *Compiler/Program Setup* section). Highlight and select the compiler interface you want to run. It will immediately start compiling

source code in the window you are editing. The output to the screen, while the compiler runs, is user-configurable. Most users, however, choose to have the screen output displayed in a pop-up window.



When finished compiling, Multi-Edit will place you on the first compiler error in the source file, assuming you have at least one compiler error. The Compiler Results are displayed in the Results Window (unless you have the **Auto-arrange Compiler Error Window** option disabled). To move to subsequent compiler errors, use **Tools | Find Next Compiler Error** or press the **Next** button in the Compiler Results Window. Instead of automatically tracking the errors as you move the cursor, the Compiler Results Window requires you to hit <Enter> or double-click. In addition, the errors in Results Window are displayed in a different color when the window comes up.

---

*NOTE: Execute Compiler can also be used to run other programs, such as a Lint utility or debugger.*

---



---

*NOTE: If you only have one compiler interface defined and you have configured Multi-Edit to bypass compiler menus with only one entry, the Select A Program list box will not appear and the compiler will run immediately. See **Tools | Customize | User Interface**.*

---



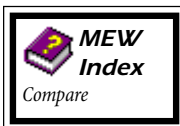
---

*NOTE: Select **Project | Options | Tools** to specify the command setups for Project release, make, build and debug tasks.*

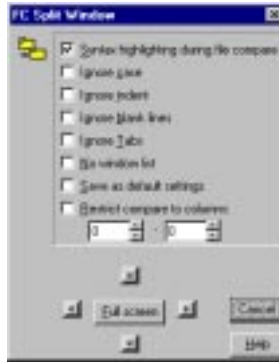
---

## COMPARING FILES

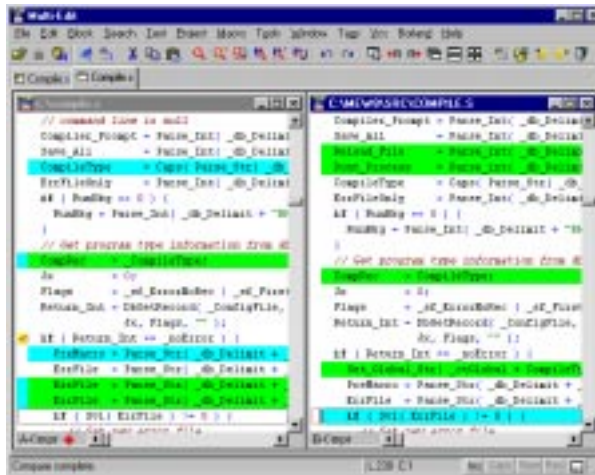
To compare two files, load them both into Multi-Edit with focus one (the current file). Default settings can be changed in **Tools | Customize | File Compare** or by checking **Save as default settings** before initiating the compare.



Select **Text | Compare files** to view the FC Split Window dialog, allowing you to make changes to the comparison criteria.



If you have selected anything other than **Prompt** in the **Window split** field, the default settings will be used and you will not see the FC Split Window dialog box, but instead be taken directly to the Link Window dialog box, a standard Multi-Edit list box that lists the open files from which to choose for the comparison. Highlight the desired file and press the **Select** button.



When the comparison is complete, the two files may be moved through and edited. The cursor movement between the two files will be linked and synchronized. By default, you may use the <Ctrl+PgUp> and <Ctrl+PgDn> keys to move quickly from difference to difference.

To stop the comparison, close either of the windows. If you select **Compare** again while either of the two comparison windows is active, the two files will be automatically re-compared. This is useful for updating the comparison after

significant changes have been made. You may also generate difference reports. You may change the file compare keys and the colors used to highlight the differences from the **Tools | Customize | Colors** dialog.

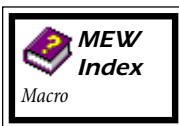


**Text | Previous Difference** and **Next Difference** are active only while performing a file compare operation. When performing a file compare, they will take the cursor to the next difference or the previous difference in the compared files.

**Text | Difference Report** allows you to generate either a detailed difference report or a summarized difference report of a file compare. The report will be written to a new window. Summarized difference reports contain the date and time the file compare was done, the names of the two files compared, and the line numbers where the two files differed. Detailed difference reports contain the same information as summarized reports, but also show the actual lines of text that differed between the two files.

**Text | Composite Diff** allows you to compare two files and build a third file that merges the differences of the two compared files into one. You can then view, save, or edit this file as necessary. This is extremely useful for merging changes to a single file from multiple sources.

## ***MULTI-EDIT MACROS***



A “macro” is a way of automating tasks, particularly repetitive tasks. Multi-Edit uses two different kinds of macros. The first, known as Keystroke Macros, is simply a sequence of keystrokes that have been recorded and may be played back later. The second is much more advanced and is created using Multi-Edit’s high level CMACW Macro Language. The Multi-Edit CMACW Macro Language is a complete programming language, designed specifically for the automation of text manipulation tasks.

A Multi-Edit macro is a sequence of instructions that the user may save, retrieve and execute repeatedly. Unlike a simple keystroke macro, which merely repeats a series of operations already performed by the user, the CMACW macro language allows for conditional action, user interface and manipulation of a variety of data types, including strings of text, integers, real numbers, and structures.



CMACW is a compiled programming language similar in syntax to C. While not as sophisticated as a full-fledged, structured programming language like C, CMACW does provide:

- ▼ if/else/else if, do/while, while, for and switch constructs
- ▼ String, integer, character and floating point data types

- ▼ Local and Global variables
- ▼ String, integer, character, real and void function types
- ▼ Parameter passing, with variable arguments and return values
- ▼ Preprocessing directives #ifdef, #ifndef, #else, #endif, #define, #undef
- ▼ Complex, nested expression evaluation
- ▼ Full access to all Multi-Edit functions
- ▼ DLL Access
- ▼ Easy access to the screen, keyboard, mouse and hardware

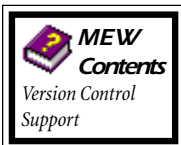
The creation of a CMACW macro is a relatively simple process: You create the source code to the macro in a Multi-Edit window and then select **Tools | Execute Compiler**. Once compiled, the macro becomes immediately available for your use. You can make changes to the macro, compile again, and the changes will instantly go into effect. If you make any syntactical errors while creating or modifying the macro, Multi-Edit will display the appropriate error message and take your cursor directly to the position where the error occurred.

For additional help with the CMACW Macro Language, select **Help | Cmac Language** or click the **CMACW Help** button on the toolbar of any Multi-Edit Help window.

The **Macro Menu** contains tools for executing and debugging Multi-Edit macros. In addition, keystroke macros may be edited from this menu.

## **VERSION CONTROL SUPPORT**

---



One of the main purposes of Version Control Support (VCS) programs is to help maintain and track versions or revisions of a set of files. Multi-Edit's VCS support consists of a set of Multi-Edit macros that acts as a front end to the more popular third-party version control systems.

The VCS support actually contains four different interfaces to the selected third-party VCS programs. This was done to provide convenience and also to help increase your productivity. With these interfaces most, if not all, of your VCS work can be done without exiting Multi-Edit.

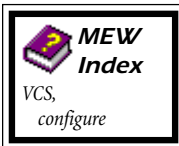
The first interface is essentially the same interface used to open and close files. When the VCS support is enabled, the file open and close functions have the ability to check files out from VCS archives when loading non-existing files, and to check files back into VCS archives when closing them or upon exiting Multi-Edit. For more detail on how these function, refer to *Using The File Open And Close Interface* in the Help System.

The other three interfaces are all accessed through the **Vcs Menu** or toolbar buttons, the most used being the *Current File Menu/Toolbar Interface*. This allows you to do most of your VCS operations using the file that is loaded into the current Editing Window.

The *Directory Of Archives Dialog* and the *Multi-Edit Files From Archives Dialog* use dialogs that allow you to pick and choose the files on which you want the VCS operations to be run. These are powerful ways to check in and out files and perform other functions in an interface similar to the Windows Explorer.

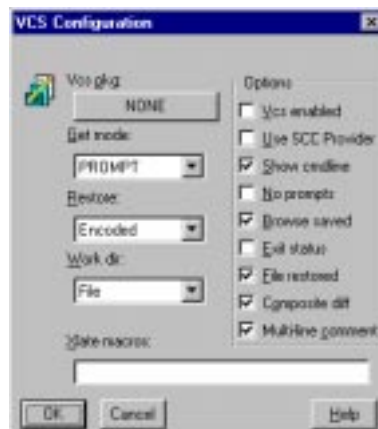
The last interface involves selecting **Vcs | Run VCS Interactively**. This starts the User Interface that comes with your VCS program without exiting Multi-Edit. This would be used for features that you require that are not currently fully supported in the VCS support macros.

## Configuring VCS

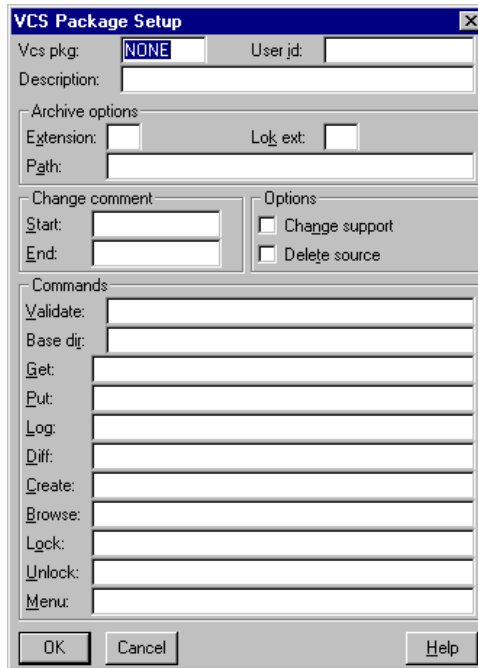


Before the VCS interfaces in Multi-Edit can be used, VCS must be configured and enabled. First, you must decide which VCS package you will be using and whether you are going to use the command line version or, if the selected package supports it, the SCC API interface. It is recommended that the SCC API interface be used since it operates much faster. Most of the supported packages have SCC interfaces but not all of them are installed by default. Check your VCS documentation to see if it is available and how to install it. If you decide to use the SCC interface, be sure that it is installed before you configure the VCS support.

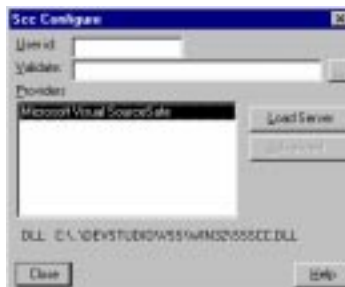
Once you have determined which VCS package you will be using, make sure the **VCS enabled** option is checked in the **Vcs | Configure** dialog. Next if you are going to be using the SCC interface be sure to check the **Use SCC Provider** option. Now select the **VCS pkg** button and you will be presented with one of two dialog depending upon whether the **Use SCC Provider** option is checked or not.



When the **Use SCC Provider** option is not checked, a dialog showing a list of all of the provided VCS packages will be shown and you can highlight the one you want to use. Selecting the **Edit** button will bring up another dialog in which you can set up the information specific to your system. The Help System provides the details about each item, but to quickly get started you should set the **Base dir** to the directory where the command line executables are installed.



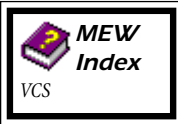
When the **Use SCC Provider** option is checked, a dialog showing a list of all of the installed SCC providers is shown. Highlight the provider that you are going to be using, and select the **Load Server** button to load the SCC provider server into memory. This only needs to be done once, as Multi-Edit will load it every time it starts as long as the VCS support is enabled. See the Help System for information on the other options.



Now, set the other options and select **OK** and, if every thing was set up correctly, the VCS support will be initialized.

One last step to be performed before the VCS support can be used is to set up the *Associate directories database*. This database is used by the VCS macros to determine where to look to find the VCS archive files when it comes time to check a file in or out.

### Using VCS



Once the VCS support is enabled and correctly configured it will run the commands needed to perform the selected function.

Checking files in and out of their archives is as easy as selecting the desired entry from the **Vcs Menu**. Multi-Edit keeps track of all of the files that you have checked out and will show you a list of these files when **Vcs | Multi-Edit files from archives** is selected. From this dialog you can check in any file that has been checked out.

You can also see the history or properties of each file by using **Vcs | History (list) current file** or **Vcs | Properties current file**.

One feature that the VCS support has added that makes your life easier is called *Change comment*. This feature allows you to add the change comments to the top of each file as you make the changes. When the file is checked back into its archive using **Vcs | Put (Update) current file menu**, the change comment that you entered at the top of the file is placed into the Describe changes edit box automatically. A change comment in a C file would appear as follows starting on line one of the file:

```
/* $change:
Any number of lines that describe the changes made to the file.
Another line of comments, etc.
$ */
```

The same comment, but in a Pascal file, would appear as such:

```
(* $change:
Any number of lines that describe the changes made to the file.
Another line of comments, etc.
$ *)
```

If a language such as Ada does not support block comments the comment would be shown as follows:

```
- $change:
- Any number of lines that describe the changes made to the file.
- Another line of comments, etc.
- $
```



# Additional Features

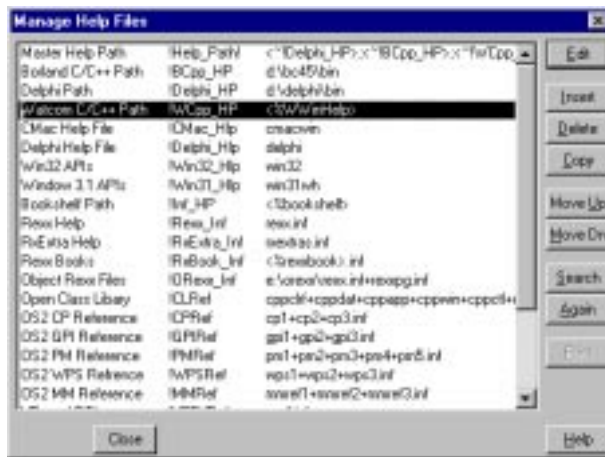
## HELP SYSTEM MANAGEMENT

### Help File Manager

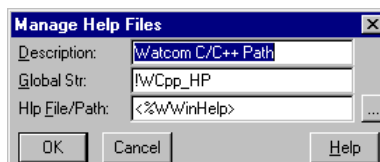


The template system allows help files to be specified for information about each template in **Tools | Edit Templates** and **Tools | Filename extensions**. To make it easier to manage all of these help files, the Help File Manager was created.

The Help File Manager creates and manages a database of help files and paths stored in the user's MECONFIG.DB. These file names and directories are assigned to global variables, which are used anywhere a help file or path would be specified. By doing this, the file name or directory only needs to be changed in the Help Manager instead of every place it might otherwise appear.



When **Help | Manage Help Path and Files** is selected, a standard Multi-Edit list box of currently defined help path and files is shown. From this dialog, new help files or paths can be defined, changed or deleted. Each line in the list displays a description of the help file or path, a global string name and the help file/paths, which may contain metacommands.



## **Viewers Setup**

To support help information other than the standard Windows .HLP format Multi-Edit has been extended to provide a user-configurable database. This database, stored in the user's MECONFIG.DB, contains the file extension of the different help formats and associated executable command lines needed to display the help data. The executables can be any type of program that Multi-Edit can launch. Select **Help | Viewers Setup** to configure.



---

## **LANGUAGE SUPPORT IN MULTI-EDIT**

---



Multi-Edit ships with support for many of the most popular programming languages (including C/C++, Pascal, Modula-2, dBase and many more) but, with a little work, unsupported languages can easily be added. Extensive support has been added for HTML, JavaScript, and other internet-related languages. These are covered in the *Programming for the Internet* section to follow.

As Multi-Edit has evolved over the years, the features that users want to change most frequently have been redesigned so that most of the changes could be made by filling in entries in dialogs, instead of writing macros. With the vast differences between programming languages, there are still some areas where macros must be written to provide the needed support.

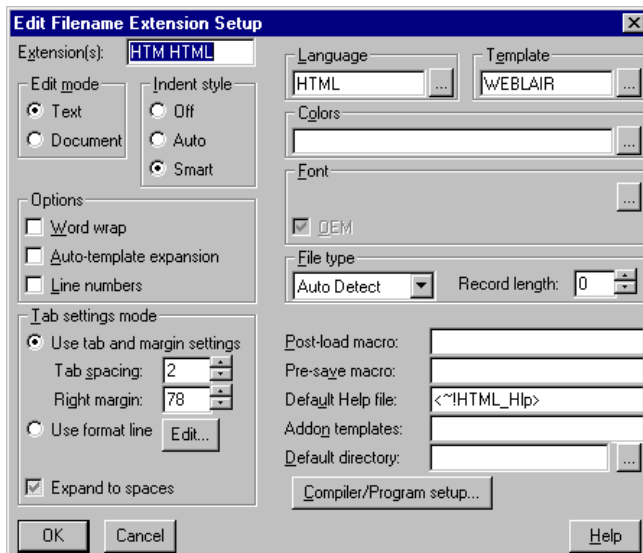
A number of different language features are provided and can be added independent of each other. When adding support for a new language, all of these features do not need to be implemented to reap the benefit of a particular feature. Thus, the new language support can be added and tested in stages.

If you have developed support for a previously unsupported language and would like it to become a supported language, send a copy of your language macro and support files (i.e., \*.TPT, etc.) and we will try to get it added to a future version of Multi-Edit.

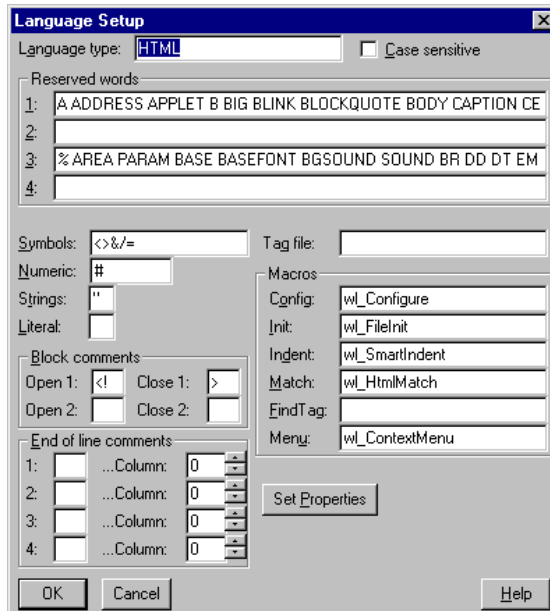
The following list of language features provided in Multi-Edit are shown from the easiest to the most difficult to implement. Not all of these features will be covered in detail here since they are covered fairly well in the Help System.

- ▼ Color syntax and keyword highlighting
- ▼ Code commenting and uncommenting
- ▼ Templates
- ▼ Compiler support and error processing
- ▼ Construct matching
- ▼ Smart indenting
- ▼ Function tagging
- ▼ Properties

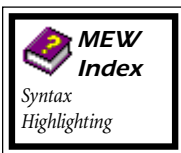
Multi-Edit supports different languages through **Tools | Filename Extensions** setup. When a file with a specific extension is loaded, a check is done to see if that extension has been defined. If it has, the language features associated with it are initialized (if not previously initialized). In the Edit Filename Extension Setup dialog, the extension-specific information is configured and the language associated with that extension is specified. In addition, the templates and compiler entries are associated with the extension.



The language-specific information is primarily defined in the Language Setup dialog (**Tools | Customize | Languages | Edit/Insert** or right click in an Editing Window and select **Customize this files settings | Language setup**). Templates are created and modified using **Tools | Edit Templates**.



### Color Syntax and Keyword Highlighting



The ability to highlight keywords and syntax with color is a very powerful feature. This and the Code commenting feature are the easiest to add. All that is required to add this feature is to create a new language record:

- ▼ Select **Tools | Customize | Languages | Insert** and type in the name of the new language. This displays the Language Setup dialog and the fields can then be filled in with the appropriate data.
- ▼ When the new language record has been defined, it must be associated with a set of file extensions in **Tools | Customize | Filename extensions**. Select a defined extension and hit **Edit** or select **Insert** and type a new set of file extensions. Select the ... button by **Language** and select the newly defined language from the list.

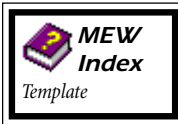
Syntax and keyword highlighting is configured and enabled in **Tools | Customize | Colors**.

## Code Commenting and Uncommenting

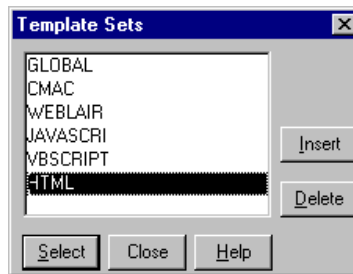


Code commenting and uncommenting is automatically added as soon as the comment fields are filled out in the Language Setup dialog.

## Templates



The new template system that is provided with Multi-Edit 8 permits the addition of abbreviations for common code fragments and language constructs that will be fully expanded when the space key is hit after typing the abbreviation. These templates are stored in \*.TPT files in the /CONFIG subdirectory. A specific language template can be defined and associated with an extension just like a language record:



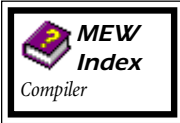
- ▼ Select **Tools | Customize | Filename extensions**. Select the ... button beside the **Template** field.
- ▼ Select a defined template and press **Select** or **Insert** to define a new language template.

There can only be one main language template associated with a set of extensions, but by adding template set names separated by ; in the **Tools | Edit templates | Addon template** field, it is possible to allow additional templates to be added to the extension. This is how a Windows API template set would be added.

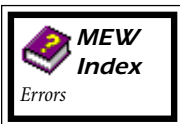
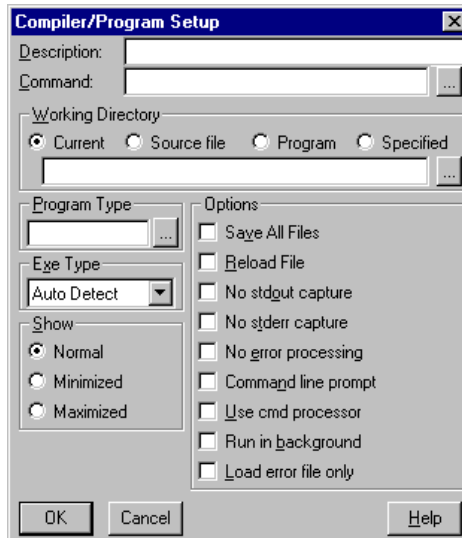
While in **Tools | Customize | Filename extensions**, be sure to check **Options | Auto template expansion** to enable template expansion for the specified extension.

To edit or add new templates, select **Tools | Edit templates**. This will open the Edit Template dialog for the language specified in the Extension setup for the currently loaded window. This is where new templates are added and older ones are changed. The fields in this dialog allow each template to be tailored for what will be inserted and when it will be expanded.

## Compiler Support and Error Processing

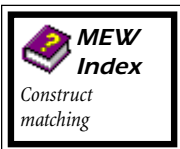


Compiler support and error processing is setup through the Edit Filename Extension Setup dialog by selecting **Tools | Customize | Filename extensions | Edit | Compiler/Program setup**. See the *Compiling Within Multi-Edit* section of this *Guide* for more details.



Adding error processing requires a little more work but it is still pretty straightforward. This involves creating a compiler type and specifying a set of regular expressions that will match the lines with errors in the captured compiler output. This is done by selecting the ... button by the **Program type** field in the Compiler/Program Setup dialog mentioned above.

## Construct Matching



Construct matching provides the ability to start with the cursor on a opening or closing construct (i.e., (), {} or begin end) and find (and optionally highlight) the matching construct. This feature can be implemented with a macro that performs the search, or through the new, general purpose LangDoMatch routine.

Once the macro is written and its name entered into **Macros | Match** field of the Language setup dialog, matching can be performed by placing your cursor on a construct to be matched (i.e., opening bracket) and select **Tools | Match Language Structure**. If a matching construct (i.e., closing bracket) is found, the cursor is positioned on it. If **Set Properties | Match language structure highlight** is enabled in the Language setup dialog, the code between the constructs is highlighted.

## Smart Indenting

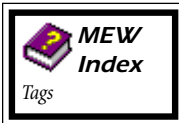
Smart indenting is the ability to position the cursor in the correct column to continue typing code after the <Enter> key is pressed. Since this feature is very language-dependent, a macro must be written which, when called by the CR system macro, should check the context of the cursor and reposition the cursor on the next line in the correct position.

---

*NOTE: Select indent style "Auto" if a language indent macro has not been written. This will cause the cursor to line up with the first word on the previous line.*

---

## Function Tagging



This feature requires a macro be written to scan the source file for function and/or variable names and writes them to a tag file in a specific format. The MeTags macro must then be patched to support the new language. This will not be covered in this *Guide*, since it is rather involved and effect system macros that constantly change. We will make this easier to add in the future.

## Properties



The Properties feature is not really a separate feature, but is used to support some of the other features. Press the **Set Properties** button in the Language Setup dialog to run the macro specified in the **Macros | Config** field. This macro should display a dialog that presents the user with configuration options that are supported in the new language support. Examples of this can be seen in the C and Pascal support where the indent style and auto highlighting of closing brackets can be enabled or changed.

If the new language is to support properties, then a set of macros needs to be written to support them. These macros are described in the Help System.

## **BORLAND IDE INTEGRATION**

---

The Borland IDE integration allows you to switch back and forth between Multi-Edit and a Borland IDE, and have both environments reflect any editing changes. It also makes available a **Borland Menu** in Multi-Edit that allows you to launch Borland IDE commands (like Compile) directly from within Multi-Edit. Currently, Delphi 1.0, 2.0, 3.0 and C++Builder 1.0 are supported. For Delphi 2.0, 3.0 and C++Builder, we allow you to synchronize IDE projects to Multi-Edit sessions.

---

*NOTE: Please review the BORLAND.TXT file (in the \BORLAND directory) and Help System for current, detailed information on Borland IDE Integration.*

---

### **Delphi 1.0**

The Delphi 1.0 integration is much more primitive than the Delphi 2.0 (and above) and C++Builder integrations. This is due to the very limited API that was exposed with Delphi 1.0. It works by saving all changed files whenever Delphi is deactivated and then, upon reactivation, checking for changes to the time/date stamps of those files and reloading the ones that have changed.

### **Delphi 2.0, Delphi 3.0 and C++Builder 1.0**

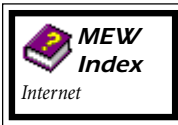
The integration with Borland's 32-bit IDEs is much more sophisticated. Most of the limitations that exist with the 1.0 integration are gone. When switching back and forth between Multi-Edit and the IDE, the actual file is not modified on disk. Instead, a temp file is written, and that is the file to which modifications are made. In Multi-Edit, the file you are editing will appear to have the same path and name as the file in the IDE, but it will actually be saved to the temp file.

---

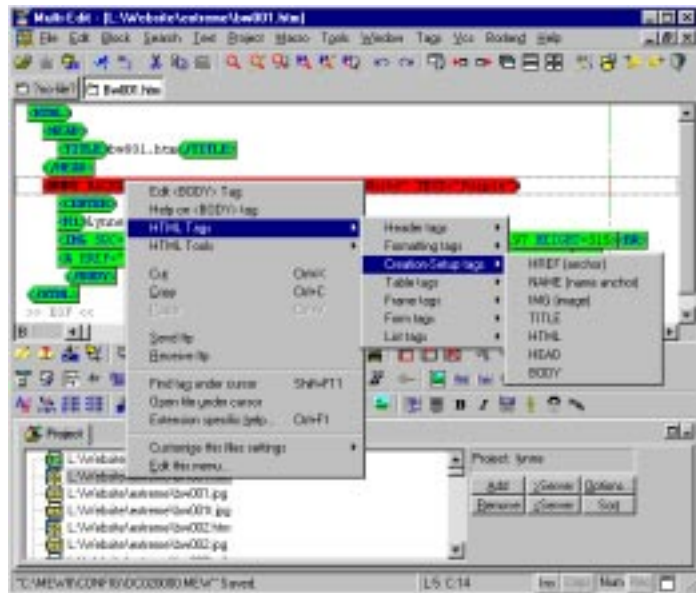
*IMPORTANT: It is the IDEs responsibility to actually save the file, overwriting the original. If you have the "Safety Autosave" feature turned on (in Multi-Edit's **Borland | Configure** dialog), then all modified files will be saved everytime you switch back to the IDE. If you do not have this option turned on, then you will have to save the files from the IDE (**File | Save**).*

---

## PROGRAMMING FOR THE INTERNET



We have expanded the language support for internet and intranet programming with WebLair, formerly sold separately as an add-on and now fully integrated into Multi-Edit 8. Internet programming language support allows for editing of HTML and Active Server Pages, including standard support for languages like Java and Perl.



When editing an HTML file, internet programming commands are accessible from the toolbars displayed below the Editing Window or by right-clicking on an Editing Window to pop-up the **Context Menu**. As with others in Multi-Edit, these toolbars and menus are fully-configurable via **Tools | Customize**. The **Context Menu** includes options for editing tags, obtaining Help on the selected tag, selection of new HTML tags, access to HTML Tools and FTP transfer of files.



## Editing Tags



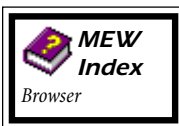
When editing an HTML file, Multi-Edit displays the HTML text with syntax-highlighted tags. SCRIPT tags are recognized automatically, showing the correct script (i.e., JavaScript or VBScript). Tags for ASP (Active Server Pages) are recognized even when embedded in a HTML tag. Each HTML tag can be edited individually by right-clicking on the tag and selecting **Edit </xxx> Tag** from the pop-up **Context Menu**.



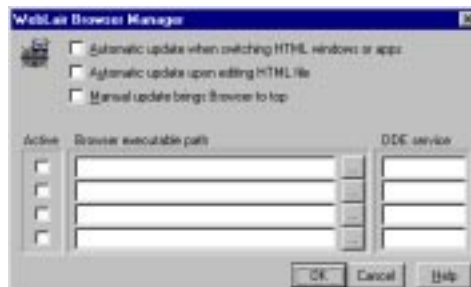
## HTML Tag Database

The HTML Tag Database can be edited, and new tags easily added by selecting **Tools | Customize | WebLair | HTML Tag Database**. This is an advanced feature and is covered in more depth in the Help System.

## Browser Configuration



Files can be viewed in Wysiwyg mode by launching your Browser from within Multi-Edit. Up to four browsers can be configured and updated at once. Configure browsers by selecting **Tools | Customize | WebLair | Browser Manager** or select **HTML Tools | Browser Manager** from the **Context Menu** in an Editing Window.



Browsers can be set to update automatically when switching to the browser or switching HTML files. To update browsers manually, select **HTML Tools | Update Browser** from the right-click menu or **Update Web Browser** from the toolbar.

### Website Management

When combined with the Project Manager, internet programming language support includes many features for managing an entire web site. One of the most useful time-saving features is the Common Code Manager, which allows you to duplicate a piece of HTML code across an entire Project list.

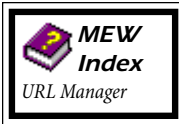


To use the Common Code Manager, mark a block of code that you will want to use in a number of places and select **Manage Common Code** from the toolbar or **HTML Tools | Common code manager** from the **Context Menu**. Select **Add** and provide a name for the block of code. This will create a marker in the HTML code and create a separate file to contain this code for later duplication and editing.

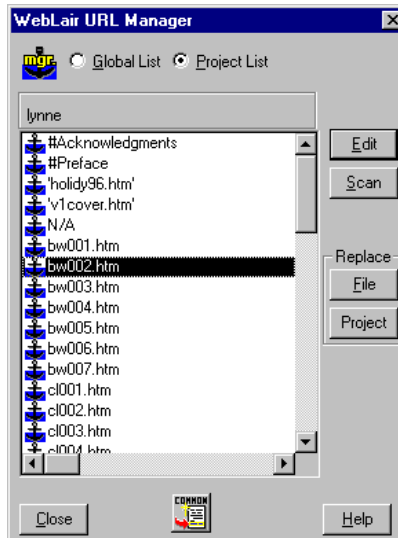


Once a common code block has been created, use the Common Code Manager to insert that block of text into each file in which you wish it to appear. This offers a great advantage over cutting and pasting or searching and replacing. When changing the common code, you can run the Common Code Manager again, edit the common code block, and update it across all files in the Project containing that common code block. This saves a significant amount of time when updating common footers and headers on your web site.

Another useful feature is the URL Manager, which performs a similar function to the Common Code Manager on URLs. You can scan your project for all URLs and, when you edit a URL, you can change it across the whole project without editing each individual file.

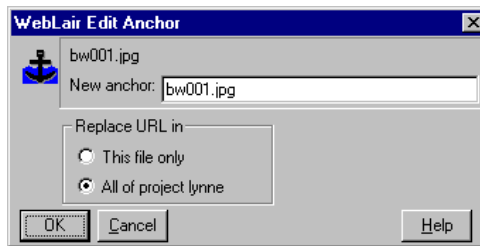


To set up the URL List, select **URL Manager** from the toolbar or **Context Menu | HTML Tools | URL manager**. Press the **Scan** button to compile a list of all URLs in all files in the Project.

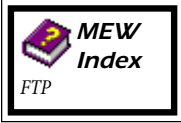


Modify a URL throughout the Project by pressing the **Edit** button. This will place a marker in the list to update this URL. When all URLs have been updated, changes can be made in the whole **Project** or just the current **File** by selecting the appropriate button in the URL Manager.

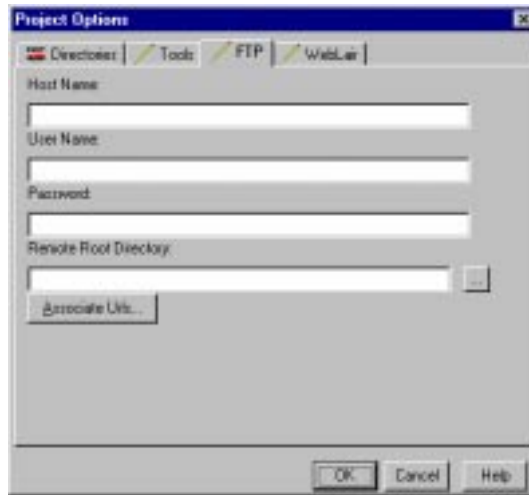
To modify a specific URL in a file, right-click on the tag and select **HTML Tools | Edit URL**. This displays the Edit Anchor dialog, in which you can modify the URL and replace the URL in the current file only or in all Project files.



## FTP



Files can be easily transferred to and from a server using the built-in FTP functions in Multi-Edit. FTP only works in combination with the Project Manager. Once you have configured a Project with the files you will be editing, select **Project | View** to view the Project Results Window. From here, select the **Project | Options** and choose the **FTP** tab.



---

*NOTE: You will also want to set up the **Directories** and **WebLair** options.*

---

From the Project Results Window, you can select files from the list on the right, then press the **>Server** or **<Server** buttons to transfer files to or from the server, respectively. Individual files can be sent or received via FTP by selecting the appropriate option from the **Project Menu** or the **Context Menu**.

---

## STARTUP OPTIONS

---



When Multi-Edit is started, the macro file STARTUP.MAC is located, searching first in the current directory, and (if not found) in the MAC subdirectory under the Multi-Edit directory. If STARTUP.MAC is found, then it is executed.

Since the Startup macro is run every time Multi-Edit is invoked, it provides a perfect place to specify any custom initialization code you may need. This also allows you to load or run macros. An example STARTUP.S macro source file is found in the /SRC subdirectory under the Multi-Edit directory. There is also a new file that uses the script setup called STARTUP.CFG. See the Help System for more information.

---

## COMMAND LINE OPTIONS

---



Multi Edit offers the ability to configure almost all of its options via **Tools | Customize**. This dialog box allows the user to set up default configurations that remain constant from one Multi-Edit session to another. Sometimes, the user needs to change a default setting temporarily, or there are some options that cannot be configured due to hardware constraints. This is where command line options can be used.

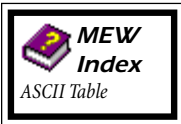
The following command line options are available:

- ▼ No Restore /NR
- ▼ No Restore, No Save /NS
- ▼ Run A Macro /R
- ▼ Display Session Manager /SM
- ▼ Start A Session /SN
- ▼ Start The Last Session /SR
- ▼ Goto Line /L
- ▼ Load A File
- ▼ Load A DOS File /\*
- ▼ Load A Unix File !/
- ▼ Load A Binary File /&
- ▼ Load a List of Files /@
- ▼ Multiple Instances /NI

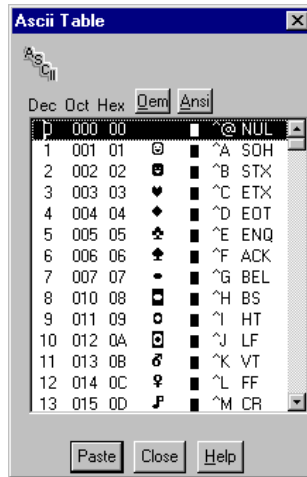
## ADDITIONAL TOOLS

These tools are conveniently built into the Multi-Edit **Tools Menu**.

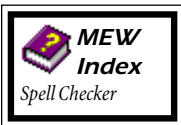
### ASCII Table



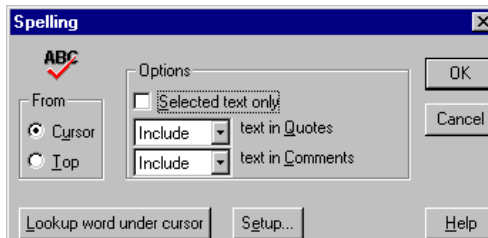
**Tools | ASCII Table** displays a dialog of all the ASCII codes from 0 to 255. Both Hex and Decimal codes are shown for two different fonts, labeled OEM and ANSI. The ASCII Table dialog is modeless (you can continue to edit text while the table still appears on your screen).



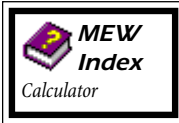
### Spell Checker



Designed to be used for memos, letters, and even string constants in your source code, **Tools | Spell Check** allows you to respond immediately to misspelled words as they are found.



## Calculator



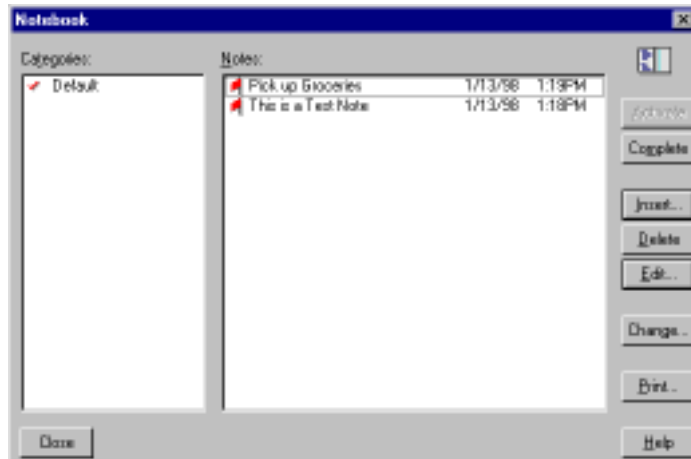
Selecting **Tools | Calculator** brings up a full-function programmer's calculator that performs decimal, hexadecimal, octal, and binary calculations. It also has an on-screen tape. The calculator is modeless, so you can keep it up and available for easy use while you work.



## Notebook



**Tools | Notebook** is a utility that allows you to compose and organize notes in a very convenient manner. When the Notebook is activated, a dialog box will appear with several buttons and two list fields.



**Categories** contains a list of your current notebook categories. This allows you to organize your notes by subject, rather than having them jumbled together in one large list. A check mark is displayed next to the currently activated category.

**Notes** contains a list of all the note titles for the currently selected category, along with the date and time they were created. Icons will appear next to the notes according to whether they are marked 'To do', 'Completed', or neither.

### ***Linedraw***



To use the line drawing feature:

- ▼ Select **Tools | Linedraw** from the Multi-Edit main menu.
- ▼ Select the type of line you want (none, single, double).
- ▼ Use the mouse to return focus to the editing window (click in the editing window).
- ▼ Move your caret (cursor) to where you want your lines to start.
- ▼ Press and hold the <Shift> key while using the arrow keys to draw the lines.

---

*NOTE: If you do not select a font within Multi-Edit that supports line drawing characters, your line drawing characters may appear incorrectly. You can set up specific fonts for each extension from the Filename Extensions dialog, and you can change the font globally from the Fonts dialog.*

---

## Symbols

---

**!** 81  
**&** 81  
**/\*** 81  
**/@** 81  
**/L** 81  
**/NI** 81  
**/NR** 81  
**/NS** 81  
**/R** 81  
**/SM** 81  
**/SN** 81  
**/SR** 81  
**<Alt+N>** 48  
**<Alt+P>** 48  
**<Alt>** 24, 41  
**<Caps Lock>** 31  
**<Ctrl+Alt+Dn>** 48  
**<Ctrl+Alt+Up>** 48  
**<Ctrl+Break>** 38  
**<Ctrl+T>** 46  
**<Ctrl+PgDn>** 60  
**<Ctrl+PgUp>** 60  
**<Ctrl+R>** 38  
**<Ctrl+U>** 46  
**<Ctrl>** 23, 48  
**<Del>** 26  
**<Enter>** 7, 34, 59, 73  
**<Esc>** 27, 31  
**<Ins>** 31  
**<Num Lock>** 31  
**<Server** 80  
**<Shift+Tab>** 41  
**<Shift>** 41, 48, 84  
**<Tab>** 41, 42  
**>Server** 80

## A

---

**Active Server Pages**  
embedded scripting 76  
language support 75  
tags 77

**Add**  
block 42  
buttons 20  
menu items 17  
file to project 55  
templates 46

**Add-on**  
templates 45

**Aliases**  
regular expression 47, 52

**ANSI** 82

**API functions** 45

**Append** 37

**Arrange**  
windows 56

**ASCII**  
codes 82  
table 82

**ASM**  
Multi-Tags 55  
tags 20, 55

**Assign**  
keys 15

**Associate directories**  
**database** 65

**Autosave** 34

**B**

---

**Backups** 34

**Binary**  
calculations 83

**Binary file type** 34, 36  
record length 34

## Block

add 42  
clear 38  
column 41  
comment 43  
common code 78  
convert 40  
copy 24, 28, 38, 40, 41  
delete 38, 41  
divide 42  
ending point 41  
extend 41  
indent 41, 42  
mark 38, 48  
math operation 42  
move 24, 38, 40, 41  
multiply 42  
non-persistent 40, 42  
persistent 40, 42  
save to disk 42  
stream 41  
subtract 42  
undent 41, 42  
window copy 38  
collapse 53  
fill 38  
fill down 38  
fill paragraph 39  
fill series down 39  
fill up 39  
menu 19, 38  
smart fill 39

**Block Operations** 38

**Bookmarks** 30, 52  
find 52  
Results Window 52  
Global 52  
search 52

**Borland**  
configure 74  
IDE Integration 7, 74  
keymap 6  
menu 19, 74  
temp files 74

**Brief keymap 6****Browse**

code 20  
Multi-Tags 55  
tags 55

**Browser**

configure 77  
launch 77  
manager 77  
update 78

**Buffer 28, 37, 42**

show 37

**Bug Reports 11****Build 59****Build Templates 46****Button**

icons 20  
style 22

**Buttons**

add 20  
create 20  
custom 22  
edit 20  
modify 20  
remove 20  
tab bar 25

---

**C****C/C++**

Builder integration 74  
Multi-Tags 55  
tags 20, 55

**Calculations**

binary 83  
decimal 83  
hexadecimal 83  
octal 83

**Calculator 83****Caps Lock status 31****Cascade**

window 35

**Case**

change 42  
proper 25

**Center**

line 44

**Change**

case 42  
comment 65

**Children**

link 35

**Classic Style dialog 33****Clear**

block 38  
random access mark 51

**Client area 27, 31****CLIPPER**

Multi-Tags 55  
tags 20, 55

**Close**

file 34  
results window 31  
window 25

**CMACW**

macro language 61  
Multi-Tags 55  
tags 20, 55

**Code**

commenting 71

**Code browser 20****Coding macro 73****Collapse 30, 53**

mode 53  
results window 53  
column 53  
expand 53  
scan 53  
search string 54  
view 53

**Color**

custom 30  
customize 25, 28, 61  
templates 28

**Column**

block 41  
comment 43  
mark 41  
collapse 53

**Command line**

options 57, 81  
session 56

**Command map 18**

command mapping dialog  
16, 18, 22  
edit 22  
mapping 13

**Command set 13**

command set manager 14  
default 19

**Comment 43**

block 43  
column 43  
line 43  
reverse 43  
stream 43  
change 65  
code 71  
commenting 43

**Common Code Manager 78****Compare**

composite difference 61  
files 59

**Compile**

macro 62  
within Multi-Edit 57

**Compiler/program**

build 59  
configure 58  
debug 59  
errors 59  
execute 58, 62  
interface 58  
make 59  
release 59  
results 30  
results window 59  
run 57  
setup 58, 72  
support 72

**Composite difference 61****Configure**

browser 77  
FTP 55  
help viewer 68  
SCC 64  
toolbar 20  
toolbox 20  
VCS 63

- compiler/program 58
- environment 57
- file compare 59
- Construct matching** 5, 72
- Context Menu**  
23, 25, 27, 75, 77, 78, 79, 80
- Copy** 37
  - block 24, 28, 38, 40, 41
  - cut 28
- CR system macro** 73
- Create**
  - buttons 20
  - toolbar 20
  - toolbox 20
  - help file database 67
  - macro 62
  - templates 46
- CUA keymap** 6
- Cursor**
  - location 56
  - move 46, 47
  - position 31
  - retrieve 47
  - save 47
- Customize**
  - buttons 22
  - command set 14
  - editing 36, 37, 42
  - file compare 59
  - filename extensions  
25, 28, 42, 45
  - files 34
  - keys/commands 15
  - languages 26
  - menus 17
  - options 19
  - tab bar 24
  - user interface 33
  - windowing 24, 25, 26, 35
  - colors 25, 28, 61
  - context menu 27
  - file settings 28
  - fonts 25
  - toolbar 20, 75
  - toolbox 20
  - language setup 70
  - sessions 56
- toolbox 75
- user interface 59
- WebLair 77
- Cut** 37
  - copy 28
- Cut and Append** 37
- D**

---

- DBASE**
  - Multi-Tags 55
  - tags 20, 55
- Debug** 59
- Decimal**
  - calculations 83
  - codes 82
- Default**
  - command set 19
  - file type 34
  - keymap 6, 17
- Define**
  - custom colors 30
- Delete**
  - block 38, 41
  - window 26
  - field marker 46
  - project 55
  - random access mark 51
  - templates 46
- Delphi**
  - integration 74
- Difference**
  - composite 61
  - next 61
  - previous 61
  - report 61
- Directory Of Archives**  
**Dialog** 63
- Display**
  - file names 25
- Display Options** 36
- Divide**
  - block 42
- Documentation headers** 45
- DOS**
  - files 34
  - load file 81
- Drag and Drop Files** 23
- Draw lines** 84
- Drop**
  - marker 51
  - random access mark 51
- Dynamic Data Exchange**  
**Help** 9
- E**

---

- Edit**
  - append 37
  - copy 37
  - cut 37
  - cut and append 37
  - file 27
  - menu 19, 37
  - menu set 18
  - paste 37
  - redo 37
  - repeat 38
  - templates  
5, 25, 28, 44, 45, 46, 67, 70, 71
  - undo 37
  - buttons 20
  - context sensitive menu 28
  - tag 77
  - templates 71
  - text 37
  - toolbox 20
  - URL 79
- Editing**
  - customize 36, 37, 42
  - windows 27
- Email**
  - technical support 10
- Embedded scripts** 76
- Enable**
  - VCS 63
- Encoded**
  - status file 56
- Environment**
  - configure 57
  - session 57
- Error**
  - compiler 59
  - processing 72

**Exclude**

hidden lines 53

**Execute**

compiler 58, 62

**Expand**

templates 45, 46, 71  
collapsed text 53

**Explicit**

tags 20, 55

**Explorer-style dialog 33****Extend**

block 41

**Extension**

setup 28, 45, 71  
specific help 28  
templates 45

---

**F**

---

**FAQs 10****FC Split Window dialog 60****Field marker 46**

delete 46  
mark 46  
strip 46

**Field strings 46**

strip 46

**File**

autosave 34  
backup 34  
close 34  
compare 59  
customize 34  
load 81  
open 26, 28, 33, 62  
prompt 33  
save 27, 34  
view 35  
customize settings 28  
add to project 55  
check in 62  
check out 62  
find 49  
information 27  
load binary 81  
load list 81  
menu 19  
merge 61

names 25

replace 50

search 49

status 56

tag 73

transfer 55

types 34

**File compare**

configure 59  
split window 60  
composite difference 61  
customize 59  
difference report 61

**File find**

dialog 49  
results window 49

**File Prompt 26, 33****File replace**

Results Window 51

**File types**

binary 34  
DOS 34  
UNIX 34

**Filename extensions  
34, 70**

compiler/program setup 57  
customize 25, 28, 42, 45, 71  
dialog 84  
help system 67  
language support 69  
project management 55  
setup 69, 72

**Files**

number 5

**Fill**

block 38  
block down 38  
block up 39  
paragraph 39  
series down 39  
smart 39

**Find**

bookmark 52  
dialog 48  
tag 28  
48  
aliases 52  
compiler errors 59

file 49

file replace 50

goto line number 52

history 56

incremental search 48

line number 52

literal string 47

next compiler error 59

regular expressions 47

replace 50

tag under cursor 54

text 47

word 48

**Find and replace 47****Flow control statements 45****Fonts**

customize 25  
ANSI 82  
ASCII Table 82  
dialog 84  
line drawing 84  
OEM 82  
style 30

**Format**

lines 44  
text 42

**FORTRAN**

tags 20, 55

**FS List 49****FTP 19, 75, 80**

configure 55  
receive file 80  
requirements 5  
results 30  
send file 80  
technical support 10

**Function tagging 73**

---

**G**

---

**Get Position**

marker stack 51

**Global**

bookmarks 52  
expression highlight 47  
session 56  
status file 56  
templates 45  
variable 22

**Go To Line** 31, 47, 52, 81  
**Goto Line Number** 52

## H

**Hardware Requirements** 5

**Headers** 45

### Help

Cmac Language 62  
 Dynamic Data Exchange Help 9  
 file manager 67  
 Macro Language Reference 9  
 WebLair HTML Reference 9  
 extension-specific 28  
 file database 67  
 file manager 67  
 manage help path and files 67  
 menu 9  
 tags 75  
 viewer setup 68

**Help System** 9

keywords 10  
 management 67

### Hex

calculations 83  
 codes 82  
 mode 36

**Hidden lines**

exclude 53  
 include 53

### Hide

window 25, 27, 35  
 results window 31  
 toolbar 20  
 toolbox 20

**Highlight**

global expression 47  
 keywords 70  
 syntax 70

**History**

find 56  
 replace 56  
 search 56

**HTML**

common code manager 78  
 embedded scripting 76  
 language support 75  
 reference guide 7  
 tag database 77

tags 75, 77  
 templates 46  
 tools 75, 78, 79  
 URL manager 79

## I

**Include**

hidden lines 53

**Incremental search**

list box 27  
 48

**Indent**

block 41, 42  
 style 73

**Information**

files 27

**Initialize**

VCS 65

**Insert mode** 31

**Installation** 6

options 7

**Integration**

Borland 74  
 C++Builder 74  
 Delphi 74

**Intellimouse support** 5, 23

**Interface**

compiler/program 58  
 user 5, 13, 61  
 VCS 62  
 vcs 63

**Internet**

technical support 10

**Invoke**

template expansion 46

## J

**Java**

language support 75  
 Multi-Tags 55  
 tags 20, 55

**JavaScript**

embedded scripting 76  
 language support 77

**Justify**

paragraph 44

## K

**Key Assignments** 5, 6, 15

**Key Mapping** 13. See also  
**Keymap**

**Keyboard**

emulation 14

**Keyboard mapping** 17

**Keyboard shortcuts** 17

**Keymap** 13

Borland 6, 14  
 Brief 6, 14  
 default 6, 17  
 Visual Studio 6  
 WordStar 6, 14  
 CUA 6

**Keys/Commands** 15

**Keystroke**

macro 31, 61  
 record 31

**Keyword**

help system 10  
 highlighting 70  
 templates 45

## L

**LangDoMatch** 72

**Language support**  
 5, 25, 28, 43, 68

active server pages 75, 76  
 coding macro 73  
 common code manager 78  
 compiler support 72  
 construct matching 72, 76  
 context menu 27  
 customize 26  
 extension specific help 28  
 syntax highlighting 30, 70  
 templates 44, 71  
 commenting 71  
 compiler 58  
 construct matching 72  
 CR system macro 73  
 debugger 58  
 edit templates 71  
 embedded scripting 76  
 embedded scripts 76  
 error processing 72

filename extensions 69, 70, 71  
function tagging 73  
HTML 75  
indent style 73  
internet 75  
java 75  
JavaScript 76, 77  
keyword highlighting 70  
language setup dialog 71  
linker 58  
MeTags macro 73  
multi-tags 20, 54, 55  
perl 75  
properties 73  
setup 28, 70, 71, 73  
smart indenting 73  
syntax highlighting 29, 76  
tags 54, 55  
template expansion 71  
template sets 45  
templates 76  
VBScript 76, 77  
VCS 65

**Language templates 45****Length**

line 5

**License Agreement 2****Line**

center 44  
comment 43  
draw 84  
drawing 84  
lengths 5  
mark 41  
number 36, 52  
numbering 36  
numbers 36  
search 52

**Link**

children 35  
window 35, 60

**List**

all occurrences 49  
Multi-Tags 54  
tags 54  
VCS files 65  
window 35  
windows 27

**List Box 27, 32, 58****Load**

file 81  
binary file 81  
DOS file 81  
files 81  
unix file 81

---

**M**

---

**Macro**

keystroke 31  
record 31  
toolbar 22  
CMACW 61  
coding macro 73  
compile 62  
ConvertBlock 40  
CR system macro 73  
create 62  
keystroke 61  
language help 7  
language indent 73  
language reference 9  
menu 19, 62  
MeTags macro 73  
properties 73  
run 81  
source files 7  
vcs 62

**Macros**

system 73

**Main Menu 19****Make 59****Manage**

common code 78  
URLs 79

**Manage Help Path and Files 67****Mark**

block 38, 41, 48  
column 41  
line 41  
starting position 41  
stream 41  
turn off 40  
drop 51  
field marker 46  
position 51

**Marker**

drop 51  
get position 51  
push position 51  
remove 51  
stack 51

**Marking Blocks 41****Math Operation**

block 42

**Maximize**

window 25, 35  
results window 31

**Menu 13, 17**

add items 17  
block 19  
Borland 19  
context  
23, 25, 27, 75, 77, 78, 79, 80  
customize 17  
edit 19, 28  
file 19  
help 9  
macro 19, 62  
main 19  
modify items 17  
project 19  
search 19, 47  
tags 20, 54  
text 19  
tools 14, 19, 82  
vcs 20, 63  
Watcom 19  
window 20

**Menus dialog 17****Merge**

files 61

**Metacommand 52**

template expansion 45

**MeTags macro 73****Microsoft Intellimouse support 23****Minimize**

window 25, 27, 35  
results window 30

**Modify**

menu items 17  
buttons 20

- toolbar 20
- toolbox 20
- Modula-2**
  - Multi-Tags 55
  - tags 20, 55
- Mouse**
  - block marking 41
  - copy block 41
  - move block 41
- Move**
  - block 24, 38, 40, 41
  - window 25, 38
  - cursor 46, 47
- Multi-Edit buffer 37, 42**
- Multi-Edit Files From Archives Dialog 63**
- Multi-Tags 54**
  - browse 55
  - explicit 55
  - find again 54
  - find tag under cursor 54
  - list 54
  - menu 20
- Multiple instances 81**
- Multiply**
  - block 42
- N**

---

- Navigate**
  - window 27
  - windows 36
- Network Version 6**
- Next**
  - window 25, 35, 36
  - difference 61
  - incremental search 48
  - template field marker 46
  - word 48
- Non-Persistent Blocks 40, 42**
- Notebook 83**
  - categories 84
  - notes 84
- Notes 84**
- Num Lock status 31**
- Number**
  - files 5
  - lines 36
- O**

---

- Octal**
  - calculations 83
- OEM 82**
- Open**
  - dialog 26
  - file 26, 28, 33, 62
  - file under cursor 33
  - window 26
  - files 56
  - options 33
- Opening Files 33**
- Options**
  - file open 33
  - user interface 33
  - command line 57
- Organizing Windows 35**
- Other templates 45**
- Overwrite mode 31**
- P**

---

- Package Contents 6**
- PARADOX**
  - Multi-Tags 55
  - tags 20, 55
- Paragraph**
  - fill 39
  - justify 44
- Pascal**
  - tags 20
- Paste 37**
  - block 28
- Patches 10**
- Perl**
  - language support 75
- Persistent Blocks 40, 42**
- Position**
  - cursor 31
  - toolbar 20
  - toolbox 20
  - mark 51
- Press Key to Assign dialog 16**
- Previous**
  - window 25, 35, 36
  - difference 61
  - incremental search 48
  - template field marker 46
  - word 48
- Primary Key 16**
- Project**
  - build 59
  - debug 59
  - delete 55
  - directories 56
  - FTP 80
  - make 59
  - manager 19, 30, 55
  - menu 19, 80
  - options 56, 59, 80
  - receive file 80
  - release 59
  - remove files 55
  - results window 55, 80
  - select 55
  - send file 80
  - set 55
  - switch 56
  - tools 59
  - view 55, 80
  - view files 55
  - website management 78
- Prompt**
  - file 33
- Proper Case 25**
- Properties**
  - language 73
  - macros 73
- Push Position**
  - marker stack 51
- Q**

---

- "Quick Pick" Window List 35**

**R****Random access mark 51**

- clear 51
- delete 51
- drop 51
- remove 51
- restore 51
- retrieve 51
- set 51

**Record**

- keystrokes 31
- macro 31

**Record length**

- binary files 34

**Redo 37****Regular expressions 72**

- aliases 47, 52
- find 47
- search 47

**Release 59****Remove**

- buttons 20
- toolbar 20
- toolbox 20
- window 26
- files from project 55
- marker 51
- random access mark 51

**Repeat 38**

- keystrokes 38

**Replace**

- file 50
- history 56
- text 49

**Resize**

- results window 31

**Restore**

- window 25, 27
- random access mark 51

**Results**

- compiler 30
- ftp 30
- search 30

**Results window 30**

- bookmark 52
- close 31
- compiler 59

- file find 49
- file replace 51
- hide 31
- maximize 31
- minimize 30
- project 55
- resize 31
- collapse 53

**Retrieve**

- cursor 47
- random access mark 51

**Reverse**

- comment 43
- undo 37

**Right mouse button 23**

- cursor position 31
- editing window 27, 70
- results window 31
- tab bar 25
- toolbar 20
- toolbox 20

**Ruler 26****Run**

- compiler 57, 59
- external program 57, 59
- VCS Interactively 63

**S****Save**

- block to disk 42
- files 34
- file 27
- cursor 47
- files 34

**Scan**

- collapse 53
- URLs 79

**SCC**

- API interface 63
- configure 64
- interface 63

**Screen**

- virtual 35

**Scripts**

- embedded 76

**Scrolling**

- Intellimouse support 23

**Search**

- aliases 52
- results 30
- list box 27
- bookmark 52
- compiler errors 59
- file 49
- file replace 50
- goto line number 52
- history 56
- incremental 48
- line number 52
- literal string 47
- menu 19, 47
- operations 47
- regular expressions 47
- replace 50
- tag under cursor 54
- text 47
- word 48

**Search and replace**

- text 47

**Secondary Key 16****Select**

- project 55
- window 24

**Session**

- bookmarks 52
- Borland 74
- buffer 37
- command line 56
- customize 56
- display 81
- environment 57
- global 56
- information 50
- management 56
- manager 52, 56
- start 57, 81
- start last 81
- working directory 56

**Set**

- project 55
- random access mark 51

**Setup**

- extensions 28
- languages 28
- options 7

**Shortcut**

keyboard 17

**Show**

buffer 37  
hidden windows 25  
minimized windows 25

**Size**

window 25

**Smart indenting 5, 73****Software Requirements 5****Spaces 42****Spacing 42****Spell Checker 82****Split**

window 35, 60

**Startup**

options 81

**Status Bar 30, 31, 38****Status file 56**

encoded 56  
path 57

**Status Line 31****Stream**

block 41  
comment 43  
mark 41

**String**

find 47  
search 47

**Strip**

field markers 46  
field strings 46

**Style**

buttons 22

**Subtract**

block 42

**Switch**

project 56

**Syntax highlighting  
26, 29, 30, 70**

embedded scripting 76  
HTML tags 77

**T****Tab bar 24**

buttons 25  
context menu 25  
customize 24

**Tabs 42****Tabs vs. Spaces 42****Tag**

browse 55  
database 54  
explicit 55  
find 28  
file 73  
find tag under cursor 54  
function 73  
HTML 75, 77  
HTML tag database 77  
list 54  
menu 20  
SCRIPT 77

**Tags**

find again 54  
menu 54

**Task List 30****Technical articles 10****Technical support 2, 10**

email 10  
web site 10

**Template**

add 46  
add-on 45  
color 28  
create 46  
delete 46  
edit 28, 44, 45, 46, 67, 70, 71  
editing 25  
embedded scripting 76  
expansion 25, 45, 46, 71  
global 45  
help system 67  
HTML 46  
keywords 45  
language 45  
metacommand 45  
other 45  
sets 45

**Text**

find 47  
find and replace 47  
menu 19, 42  
replace 49  
search 47  
search and replace 47

**Tile**

window 35

**Toolbar 13. See also****Toolbox**

configure 20  
create 20  
customize 20, 75  
hide 20  
modify 20  
position 20  
remove 20

**Toolbox 13, 20. See also****Toolbar**

configure 20  
create 20  
customize 20, 75  
hide 20  
manager 20  
modify 20  
position 20  
remove 20

**Tools Menu 14, 19, 43, 82****Track revisions 20****Track versions 20****Transfer**

files 55

**Turbo-Pascal**

Multi-Tags 55  
tags 20, 55

**U****Uncomment 43****Undent**

block 41, 42

**Undo 37****Unhide**

window 35

**UNIX**

file type 34  
load file 81

**Update**

browser 78

**Updates 10****URL List 79****URL Manager 79****User Interface****5, 13, 33, 61**customize 33, 59  
options 33  
vcs 63

---

**V****VBScript**embedded scripting 76  
language support 77**VCS 62**archive 62  
archive files 65  
associate directories database 65  
change comment 65  
check in file 62  
check out file 62  
close file 62  
configure 63  
configure SCC 64  
current file menu/toolbar  
  interface 63  
directory of archives dialog 63  
enable 63  
file open and close interface 62  
history 65  
history (list) current file 65  
initialize 65  
interface 62  
language support 65  
list files 65  
load server 64  
macro 62  
menu 20, 63  
Multi-Edit files from archives  
  dialog 63  
open file 62  
package setup 64  
properties 65  
properties current file 65  
put (update) current file menu 65  
run VCS interactively 63  
SCC API interface 63

SCC configure 64

SCC interface 63

usage 65

use SCC provider 63

**Version Control Support  
20, 62. See also VCS****View**file 35  
collapsed text 53  
files in project 55  
project 55**Viewer Setup 68****Virtual screen 35****Visual Studio keymap 6**

---

**W****Warranty 2****Watcom Integration 7****Watcom Menu 19****Web site**

technical support 10

**WebLair 75**Common Code Manager 78  
customize 77**WebLair HTML Reference 9****Window**arrange 56  
cascade 35  
copy 38  
hide 25, 27, 35  
link 35, 60  
list 26, 27, 35  
move 25, 38  
next 25, 35, 36  
previous 25, 35, 36  
select 24  
split 35, 60  
tile 35  
unhide 35  
zoom 35  
close 25  
delete 26  
maximize 25, 35  
menu 20  
minimize 25, 27, 35  
navigate 27  
open 26

remove 26

restore 25, 27

size 25

menu 35

move 38

single window mode 35

**Windowing**customize 24, 25, 26, 35  
single window mode 35**Windows**

clipboard 37

**Word**

find 48

search 48

**WordStar keymap 6****Working directory 56**

---

**X****XBASE**Multi-Tags 55  
tags 20, 55

---

**Z****Zoom**

window 35



