

CodeWright 6.5

Getting Started

Implementation

protected:

virtual ~CCWtxtVw();

#ifdef DEBUG

virtual void TraceValid();

virtual void DumpContent();

#endif

CLASSES

edit

METHODS

code

Update()

PROGRAMMING

```
<html>
<html>
<heading>
<heading>
<heading>
<heading>
<body bgcolor=55ff11>
<body bgcolor=55ff11>
</body>
</html>
```

TEAM

objects

JAVA

PYTHON

PERL

C++

HTML

VB

XML

```
String rs;

cellColor = Col;
inputColor = new
inputFont = new
titleFont = new
title = getParam
if (title == nul
title = "Spr
)
rs = getParamete
if (rs == null)
if (rs == null)
rs = getParamete
if (rs == null)
columns = 5;
) else {
columns = In
)
cells = new Cell
char l[] = new c
for (int i=0; i
for (int j=0; j
cells[i][j] = ne
Color.lig
Color.bis
cellColor
cellFont
cellTitle
Dimension d = s;
inputArea = new
inputColor, Col
resize(columns +
(rows +
)
public void
if (selectedRow
return;
cells[selectedRc
repaint();
)
public void
isStopped = fal;
isStopped = fal;
public void
int i=0; i
for (int j=0; j
if (cells[i][j]
)
)
public void
if (selectedRow
return;
cells[selectedRc
repaint();
)
public void
int cx, cy;
if (! fullUpdate
cx, cy;
g.setFont(t);
for (int i=0; i
for (int j=0; j
if (cells[i][j]
cx = (j * cellW;
cy = ((i+1) * ce
cells[i][j].pair
)
)
) else {
print(g);
fullUpdate;
```

Copyright © 1991 – 2000 Starbase Corporation. All rights reserved. Premia Corporation is now owned by Starbase Corporation.

The following copyright message is required due to the inclusion of CTL3D.DLL with our product:

Portions © Microsoft Corporation, 1985-1995. All rights reserved.

Publication History

November, 1995	Initial Release
July, 1996	Additions and corrections
January, 1997	Updated for version 5.0
January, 1998	Updated for version 5.1
May, 1999	Updated for version 6.0
June, 1999	Minor corrections
July, 1999	Minor corrections
October, 1999	Updated for 6.0b
February, 2000	Updated for 6.0d
Aug, 2000	Updated for 6.5
December 2000	Updated for 6.5b

Trademarks

Starbase® and CodeWright® are registered trademarks of Starbase Corporation.

CodeSense™, CodeFolio™, ChromaCode™ and ChromaCoding™ are trademarks of Starbase Corporation.

Code Composer Studio™ and Texas Instruments™ are trademarks of Texas Instruments Incorporated. TICCSync only works with Code Composer Studio™ from Texas Instruments™.

Adobe® and Acrobat® are registered trademarks of Adobe Systems Incorporated.

Borland C++®, Borland C++ Builder®, BRIEF®, and Delphi® are registered trademarks of Borland.com, a division of Inprise Corporation.

Microsoft® is a registered trademark of Microsoft Corporation.

Windows™, Visual C++™, and Visual Basic™ are trademarks of Microsoft Corporation.

Epsilon™ is a trademark of Lugaru, Inc.

AppBasic uses WinWrap® Basic, Copyright © 1993-1996, Polar Engineering and Consulting.

Sentry Spelling-Checker Engine, Copyright © 1993-1997, Wintertree Software, Inc.

Contact Information

Starbase Corporation
9615 SW Allen Blvd., Suite 107
Beaverton, Oregon 97005
Phone: (503) 641-6000; Fax: (503) 641-6001
Email: codesales@starbase.com or codesupport@starbase.com
World Wide Web: <http://www.starbase.com>

Table of Contents

Publication History	ii
Trademarks	ii
Contact Information	ii
TABLE OF CONTENTS	III
<hr/>	
INTRODUCTION	1
<hr/>	
System Requirements	1
CodeWright Specifications	2
File Size	2
Line Length Limit	2
Lines per Buffer Limit	2
Buffers Limit	2
Windows Limit	2
Number of Files	2
Clipboard/Scrap Buffer Size	2
Search String Length	2
Additional Product Support	3
Web Page	3
Internet Mail	3
Fax	3
Phone Support	3
User's Guide	3
INSTALLATION PROCEDURES	5
<hr/>	
Network Considerations	5
Upgrading from Previous Versions	5
Merging Changed Source Files	6
Your Configuration Files	7
Running Setup	7
License Agreement	7
Serial Number	7
Installation Directory	8
Installation Options	9
Selecting Your Initial Keymap Command Set	10
Selecting Synchronization Environments	10
CodeWright Program Group	10
Installing CodeSense Libraries	11
Finishing Up	11
Other Setup Considerations	11
FTEE Utility	12

Environment Variables	12
Multiple Projects and Configurations	14
Network Installations	15
RUN CODEWRIGHT FOR THE FIRST TIME	17
<hr/>	
Configuration Wizards	17
A First Look	18
The Menu Bar	19
File Menu	19
Edit Menu	20
Search Menu	21
Project Menu	21
Text Menu	22
Document Menu	22
Customize Menu	23
Tools Menu	25
Window Menu	26
The Difference between Windows and Documents	26
Details of the Window Menu	27
Help Menu	27
Output and Project Windows	28
Output Window	28
The Project Window	29
More on the Open Tab	30
The Standard Toolbar	31
INDEX	33
<hr/>	

Chapter 1

Introduction

CodeWright is an editor with features geared towards software programmers. It is designed to support multiple programming languages in that (at a minimum) it ChromaCodes™ the syntax elements of the language. It is the first professional-quality, extensible programmer's editor that has been written for Windows from the ground up.

Use this manual to install CodeWright on your system and begin various setup and configuration tasks.

System Requirements

Refer to the following system requirements for running CodeWright:

- *System:* For CodeWright 6.5 a 486-class machine is required as a minimum, with at least 16MB of RAM.
- Performance will vary depending on the video adapter, driver and video mode you have selected.
- *Memory:* CodeWright requires 1MB of memory to operate.
- *Windows:* You must be running Windows 9x, 2000, ME, or Windows NT 4.0 or greater to run the 32-bit version of CodeWright.
- *Storage:* A CD-ROM drive and a hard drive are required. CodeWright will store temporary files on the hard drive, or other location specified for temporary files. The amount of storage space required varies according to the size and number of files that are being edited.

CodeWright Specifications

CodeWright has the following capabilities:

File Size

The Default file size is 500 Mb. Using the **-BlockSize=** command line flag described in the *User's Guide* chapter on *Large Files*, a single 2Gb file can be edited (or two 1Gb files). The default block size (0x2000) allows a single 500-Mb file to be opened and edited in CodeWright. Even larger blocks would allow even larger files, but 2Gb will still be the overall limit.

Line Length Limit

Line size can be as large as 2Gb when the **-Blocksize** parameter, mentioned above, is used.

Lines per Buffer Limit

Using the **-Blocksize** parameter, the number of lines per buffer can be as many as comprise a 2Gb file.

Buffers Limit

The maximum number of buffers is limited only by storage (memory and disk).

Windows Limit

Microsoft Windows limits the number of windows.

Number of Files

The number of files being edited is limited to 65,535, subject to the limit on total file size given previously.

Clipboard/Scrap Buffer Size

There is no limit to the size of the Windows Clipboard. Also, CodeWright allows multiple Clipboards to be used. Alternatively, multiple Scrap buffers can be used, operating under the buffer limits described above.

Search String Length

Search strings are limited to 64K.

Additional Product Support

If a problem arises in using or programming CodeWright, there are a number of resources you can use. First, check the README.TXT file that was shipped along with CodeWright. It is placed in CodeWright's home directory during installation. It will warn you about any known "gotchas" that are not covered by the manuals. The next places to look for help are in this manual, the User's Guide, and the online help files.

Web Page

You can access our home page on the Internet's World Wide Web using the following URL: <http://www.starbase.com>. Information and services available there include:

- messaging to sales and product support
- pricing and product descriptions
- problem and enhancement report forms
- downloading of Add-Ons and patches

Internet Mail

You can send email to codesupport@starbase.com, and we will assist you as soon as possible. For sales matters, use the address codesales@starbase.com.

Fax

A fax will normally get a quick response. Our fax number is (503) 641-6001.

Phone Support

If you urgently need some information in order to continue using CodeWright, or if you have an urgent bug report, give us a call. The phone number is (503) 207-1190.

User's Guide

The *User's Guide* is referred to throughout this manual. If you would like to access an electronic copy, it is available in .pdf format when you insert the CodeWright CD; choose **Documentation** followed by **User Guide**. The manual displays automatically in Adobe Acrobat Reader. Acrobat Reader must be installed locally for this to work.

You can also purchase additional hard copies of the *User's Guide*. Contact your sales representative by telephone, or at the following Email address:

codesales@starbase.com

Chapter 2

Installation Procedures

CodeWright has the flexibility to support a wide variety of installations, and much of this section is devoted to presenting the alternatives available to you.

The CodeWright installation is simple. Normally, it does not require modifying your CONFIG.SYS file or your AUTOEXEC.BAT files.

Network Considerations

If you are considering installing CodeWright on a network, you should note several things. Each user needs to maintain some private files for CodeWright's use. These files nominally include a configuration file and a state file. Environment variables may be used to point to these private files.

In addition, users may wish to make their own modifications and extensions to the CodeWright executables. This would require them to have a private copy of the source code files and DLL for each portion of CodeWright they plan to change. When this is the case, the savings in disk storage gained by placing the few, shared files on the network are negligible. It is therefore usually simpler to install a separate copy on each machine. This ensures that no one will accidentally destroy another person's customizations.

A suggested setup for servicing users from a network copy of CodeWright is outlined later in this manual.

Upgrading from Previous Versions

If you are upgrading CodeWright from a previous version, you may find it easiest to install the new version to a new directory, if you can spare the disk space. You can later uninstall (using Add/Remove Programs in Win9x/2000/ME/NT's Control Panel) and delete the old files and directory when your transition to the new version is complete. The disadvantage is that the setup program will not be able to transport your old settings from your old configuration file. However, this can be done manually by running a utility called UPDATINI.EXE. UPDATINI is a DOS utility located in the CodeWright installation directory.

In DOS, run UPDATINI with the old and new configuration files in the following way:

```
updatini <old.ini file> <new.ini file> [<output.ini file>]
```

Where:

<old.ini file> is the path and name of an old CodeWright .INI file.

<new.ini file> is the path and name of the new CodeWright .INI file.

[<output .ini file>] is the path and name of the updated/merged output .INI file. Default output file is CWRIGHT.INI in the current directory.

Notes: Newer versions of CodeWright run UPDATINI.EXE automatically when upgrading existing installations (i.e. installing over old copies), thereby rendering the above instructions unnecessary.

The name of CodeWright's configuration file is CWRIGHT.INI, located by default in the CodeWright installation directory.

If updating an existing CodeWright installation, be sure to backup any CodeWright source files that may have been changed since the previous installation. CodeWright source files are supplied for users who intend to modify the way CodeWright works by rewriting the program. These source files are only installed when the **Full** option is chosen at installation. Any existing CodeWright source files will be overwritten, if it is chosen that they be re-installed.

Also, make sure that there are no read-only files in the directories. Such files typically result from changing a source code file and checking it into version control. Attempting to overwrite a read-only file will cause the installation to abort.

Whether you install to a new directory or update the old one, be sure to read the file README.TXT when installation is completed.

Merging Changed Source Files

You can use the supplied **AutoMerge** utility to combine the changes you made to CodeWright's source code with the new version of the source code files. This utility is described and supplied in the SUPPORT\AUTOMRG directory of the CodeWright CD-ROM.

Your Configuration Files

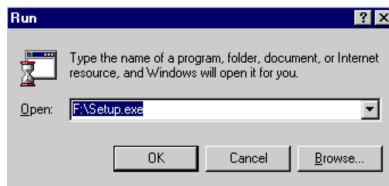
The CodeWright installation program does not overwrite your configuration files (CWRIGHT.INI and CWRIGHT.PST). Instead, it first makes a copy of the existing files to a subdirectory named PREVIOUS. It then modifies the original files to work with the latest version of CodeWright. A generic, bare bones configuration file is also supplied for reference. It is named CWRIGHT.I65. CWRIGHT.I65 is stored in the DEFAULTS subdirectory of the CodeWright installation directory.

Running Setup

CodeWright is provided on CD-ROM and supports the Autorun feature of Windows 9x/2000/ME/NT. This means that on most machines running these platforms, the Setup program will start up automatically when you insert the CD-ROM into the drive. Select from the options provided to start the CodeWright installation.

Note: To manually begin the installation program, place the CD into your CD-ROM drive. Then press the **Start** button and click **Run**.

If your CD-ROM is designated drive F, issue the command depicted below. If it is designated as another drive, just substitute that drive in the following command:



License Agreement

You must agree to the screen that contains Starbase's license agreement for CodeWright in order to continue with the installation. A printed copy of the agreement is provided along with your CD in the original package.

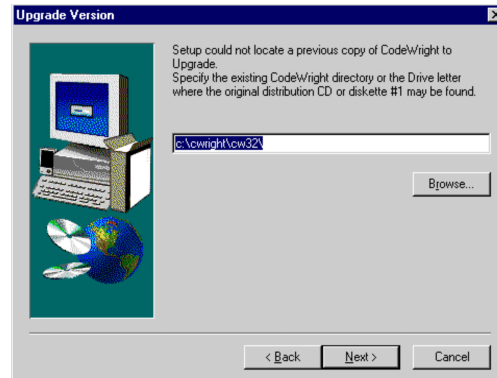
Serial Number

The next screen prompts you for a serial number. The serial number is usually located on the license agreement and/or on the back of the CD sleeve. You must insert the serial number to continue the installation. Keep in mind the following when inserting serial numbers:

- **Installing for the first time**—With initial installations, simply locate the serial number on the license agreement and/or on the back of the CD sleeve, and insert it at the prompt.

- **Upgrades**—With upgrades, the setup script will attempt to locate a previous installation of CodeWright after the serial number is inserted. If no previous installation exists, the following dialog displays:

CodeWright Program Location



If the **Upgrade Version** dialog appears, do one of the following:

- ✓ Specify the location of a local or network installation of the original version of CodeWright (i.e. the version being upgraded), if one exists. Then click **Next**. OR
- ✓ Insert the original version of the CodeWright CD, or disk 1 of the floppy diskettes, and use the above dialog to specify the location of the disk (usually the local CD-ROM drive or the A:\ drive). Once the original version has been verified, reinsert the upgrade disk to continue.

Note: If **Auto Insert Notification** is enabled for your CD ROM drive, hold down **SHIFT** while changing CDs. Otherwise, click **Cancel/Exit** to keep other install programs from automatically starting when the CD is inserted.

If problems arise, contact Starbase Product Support.

Installation Directory

At some point the program will prompt you to select an installation directory. The CodeWright installation program allows you to specify a long filename for CodeWright's installation directory, if desired.

Installation Options

The installation program lets you select from several predefined installation packages:

- *Typical*: This installation package gives you the complete installation, with the exception of CodeWright's DLL source code files. Source code files are made available with full installations of CodeWright for users who intend to change the way CodeWright works by modifying its source code.
- *Compact*: This setup is the bare minimum required to run CodeWright. Select this for laptops and anywhere storage space is in short supply.
- *Full*: Possibly as popular as the *Typical* option above, this installation package installs everything a user normally needs, including source code. If you have enough storage space, this option is recommended.

When you choose the *Full* option, a **Compiler Options** dialog displays. The compiler option you choose now will cause the installation program to install the appropriate source files needed for recompiling CodeWright DLLs. Select one of the following:

- ✓ *Microsoft*: If you intend to use Microsoft's compiler to recompile CodeWright DLLs.
- ✓ *Borland*: If you intend to use Borland's compiler to recompile CodeWright DLLs.
- ✓ *Other*: If you intend to use a compiler other than Microsoft or Borland to recompile CodeWright DLLs.
- *Custom*: This setup allows you to design your own installation. You may install just the executables, both the executables and the C language source code for the DLLs, or just the C source code. This option is also useful when you are adding files to an existing installation.

The executables by themselves (*Compact*) require about 10 MB of disk storage. A *Typical* installation uses 36 MB, while a *Full* installation uses 49 MB (without CodeSense Libraries).

If you install only the executables, you will be able to configure CodeWright just as you would if you had installed the complete package. The ability to modify or extend CodeWright will be severely hampered, however. If you elect to install only the executables and later find that you want the complete package, you may run the Setup program again, choose *Custom*, and install only the source code.

Selecting Your Initial Keymap Command Set

After choosing the type of installation you want, you will be prompted to choose your initial keymap command set. Choose from:

- CUA command set
- CodeWright 4.0 CUA (emulates the CUA keymap used in previous versions of CodeWright)
- BRIEF command set (emulates the BRIEF editor, originally from Underware)
- vi command set (emulates the vi editor found on Unix systems)
- Epsilon command set (emulates the Emacs-derived Epsilon editor from Lugaru)

Selecting Synchronization Environments

Once the keymap choice has been made, a window appears presenting you with a list of Synchronization programs that can be installed on your system. Synchronization allows CodeWright to be used in conjunction with certain GUI-based development environments. See the *User's Guide* chapter on *Synchronization* for more information.

The choices of environments to synchronize with are as follows:

- Microsoft Visual Basic (V6.0)
- Microsoft Visual Studio (V5.x or V6.x).
- Borland C++ (V5.0)
- Borland C++ Builder (V4.0 and V5.0)
- Borland Delphi 32 (V4.0 and V5.0)
- Texas Instruments Code Composer (V1.0 and V1.1)

It will be necessary to perform custom installations later to install any Synchronization programs that are not installed at this time.

CodeWright Program Group

The final screen of the installation program gives you the option of approving the CodeWright program group, which will contain the shortcuts for starting the various programs necessary for using CodeWright. If it is okay to have the shortcuts installed in the CodeWright folder, click **Next>**. Otherwise, change the name of the folder or choose **Skip This>**. The CodeWright installation will begin.

Installing CodeSense Libraries

After CodeWright is installed, a screen with important information about the install will display. Click **Next** to display a screen that optionally installs CodeWright's default CodeSense libraries. If you choose to install the default CodeSense libraries, a second install program will begin.

The first screen in the CodeSense Libraries installation program prompts for a location in which to install the libraries. The CodeWright home directory is used by default.

The next screen displays a list of CodeSense libraries. All libraries are selected by default. Select only those that you wish to install. Items selected in this screen will be added to the **CodeSense Global Configuration** dialog.

Finishing Up

Once CodeWright and the CodeSense libraries have been installed, you can invoke CodeWright to display the following files:

- *README.TXT*: Contains information too late for inclusion in the manuals.
- *UPDATE.TXT*: Contains useful upgrading information.

If you choose not to install the CodeSense libraries, you can launch CodeWright from the program group. To access the program group, press the Windows **Start** button, then **Programs**, then **CodeWright**.

Other Setup Considerations

For many users, the above installation procedure is all that is necessary to begin using CodeWright. Take a few moments, however, to look over the following information to see if your configuration could benefit from a little fine-tuning. Topics include:

- FTEE Utility
- Environment Variables
- Multiple Projects and Configurations
- Network Installations

FTEE Utility

As part of installation, a DOS program named FTEE.EXE is installed into the CodeWright home directory (FTEE32.EXE on Windows NT). This program dynamically splits output so that output from a command can be viewed, even if that output is being redirected to a file. Splitting output is especially useful when compiling or building an executable from within CodeWright, but may also be useful for purposes unrelated to CodeWright. We refer to FTEE.EXE as FTEE.

Most pre-set CodeWright command lines use the macro `${FTEE}`, which expands to FTEE.EXE (Win 9x/2000/ME) or FTEE32.EXE (Win NT) when launched. VDOS, an alternative to FTEE, is also offered. VDOS runs in a child Window and makes FTEE unnecessary. Both are discussed in the *User's Guide* chapter entitled *Set up a Compiler*.

Environment Variables

Using CodeWright does not require that you create any new environment variables. However, using environment variables to tell CodeWright where certain files are adds flexibility to your configuration. In network installations, environment variables are highly recommended for pointing to the location of private files.

CodeWright environment variables can be created and used in very flexible ways:

Example: Using `Set FOO=Zippy` in the operating system's environment causes CodeWright to substitute Zippy when `${FOO}` is encountered in any CodeWright command line.

Established CodeWright environment variables are listed in the following table.

Environment Variables	
Name	Meaning
TMP, TEMP, CWTMP	Names the directory in which temporary and "swap" files are placed. If it is not defined, these files are created in the working directory. Windows uses the TEMP variable for locating a place for its temporary files. If you want CodeWright to use a different location, use the TMP or CWTMP variable to define that location.
CWINI	The directory, or the directory and filename, where the configuration file is found. Use it only when the configuration file is not in CodeWright's home directory, the working directory or the Windows directory (e.g., network installations). Do not define this variable if you intend on using multiple configuration files.
CWLEX	The directory, or the directory and filename, where the file containing ChromaCoding lexer configuration information is stored.

Environment Variables	
Name	Meaning
CWLIB	<p>A list of directories in which Dynamic Link Libraries (DLLs) needed by CodeWright may be found:</p> <ul style="list-style-type: none"> ■ If the libraries are in CodeWright's home directory or the working directory, this variable is not necessary. ■ If you are installed on a network and plan to modify the supplied DLLs, you will probably want to use this variable. Without it, CodeWright looks only in the directory that contains its executables. <p>Since this environment variable specifies a list of paths, you can use it to mix a combination of public and private DLLs (separated by semicolons).</p> <p>Example: If the following environment variable is set:</p> <p>Set CWLIB=c:\cwlocal;k:\cwnet\DLLs;k:\cwnet</p> <p>CodeWright will look first in the local directory for DLLs, then in a special subdirectory on the network, and finally in its home directory.</p>
CWMENU	<p>The directory or directory and filename where the Popup Menu Definition file is located. If this variable is not defined, CodeWright will look for the file CWRIGHTMENU in the directory containing its executables.</p>
CWPST	<p>The directory or the directory and filename where CodeWright's state file is found. Use this only when the state file is not in CodeWright's home directory, or the working directory (e.g., network installations). Do not use this variable if you plan on using multiple state files.</p>
CWSENSEINI	<p>The full path to CWSSENSE.INI, the file that contains CodeSense configuration data. This path defaults to <CodeWright directory>\CWSSENSE.INI.</p> <p>The CWSSENSEINI variable allows you to share CWSSENSE.INI options on a network.</p>

Environment Variables	
Name	Meaning
CWSENSELIBDIR	<p>The directory in which the user's non-project-specific CodeSense library databases reside. This defaults to <CodeWright directory>\SenseDBs. The CWSENSELIBDIR variable allows you to share CodeSense library databases on a network.</p> <p>Alternatively, the [Options] section of CWSENSE.INI contains CWSenseLibDir, a value that is used just like the CWSENSELIBDIR environment variable and defaults to <CodeWright directory>\SenseDBs.</p> <p>Exercise caution, however, as sharing CodeSense library databases on a network has previously been known to cause instability in the CodeSense database.</p>

At installation, the configuration file and state file are placed in the CodeWright home directory. The Popup Menu Definition file is also placed in the home directory. If you create environment variables to point to other locations for these files, you can move them from the CodeWright home directory to their new locations. The names of these files are CWRIGHT.INI for the configuration file, CWRIGHT.PST for the state file, and CWRIGHT.MNU for the pop-up menu definition file.

As an alternative, you can specify a configuration file and a state file for CodeWright to use via its command line. You can read more about CodeWright's configuration files and command line options in the *User's Guide* chapter entitled *Configuration Files & Command Line Parameters*. The procedure for editing popup menus is described in the *User's Guide* chapter on *Custom Interface*.

Multiple Projects and Configurations

When launching CodeWright, you may wish to select from different CodeWright configurations. You may want CodeWright to use different working directories for different projects or languages. You can do this in one of two ways:

- Use multiple configuration files and program items (icons/shortcuts) for each anticipated purpose. This may be desirable if you need to invoke CodeWright with different command line options.
- Use CodeWright's Project Management features. This option is usually preferable, since you can change files and configuration on the fly.

CodeWright projects are covered in greater detail in the *User's Guide* chapter entitled *Projects, Project Spaces, and Workspaces*. Descriptions of CodeWright's command line parameters are available in the *User's Guide* chapter on *Configuration Files & Command Line Parameters*.

Network Installations

If you have installed CodeWright on a network, you will want to make a local copy of the configuration and state files:

- A local copy of the configuration file allows your CodeWright setup to be different than the setup other users select.
- A private copy of your state file contains information about the state of CodeWright when you last exited it. This allows that state to be restored when next you run CodeWright.

For most users, the simplest method for setting this up is to create the environment variables CWINI and CWPST. Let us say, for example, that CodeWright is installed in K:\CWNET, and that you are going to keep your local files in C:\CWLOCAL. These two environment variables would be defined as follows:

```
set CWINI=c:\cwlocal
set CWPST=c:\cwlocal
```

Note: It is best to remove the configuration and state files from the network directory where the CodeWright executables have been installed. That way, you will receive an error message if the intended files are not found. If you do use the configuration or state file on the network, someone else may overwrite it, or you may be unable to change settings, if you have insufficient rights in that directory.

Chapter 3

Run CodeWright for the First Time

This section describes the way CodeWright looks when it is first loaded, giving some tips about things to know before getting started. Many of the items described will be deferred to chapters in the *User's Guide* for more information. A copy of the *User's Guide* is available on the CodeWright CD in PDF format. The descriptions provided in this chapter are brief, intended only to familiarize the user with the initial CodeWright environment.

Configuration Wizards

If the option to load CodeWright after installation is chosen, the initial CodeWright screen will display with the **Wizard Choices** on top. Configuration Wizards assist with various jobs in CodeWright. The **Wizard Choices** include:

- The **Answer Wizard**: Displays lists of help topics in response to questions you enter.
- The **Help Index File Wizard**: Goes through steps for configuring the help index file. For more information about configuring help, see the topic *Using Help in CodeWright* in the *User's Guide* chapter on *Editing & Printing*.
- The **Sync Technology Wizard**: Goes through the steps needed to configure CodeWright to synchronize with selected development environments. Supported environments are:
 - ✓ Microsoft Visual Basic (V6.0)
 - ✓ Microsoft Visual Studio (V5.x or V6.x).
 - ✓ Borland C++ (V5.0)
 - ✓ Borland C++ Builder (V4.0 and V5.0)
 - ✓ Borland Delphi 32 (V4.0 and V5.0)
 - ✓ Texas Instruments Code Composer Studio (V1.0 and V1.1)

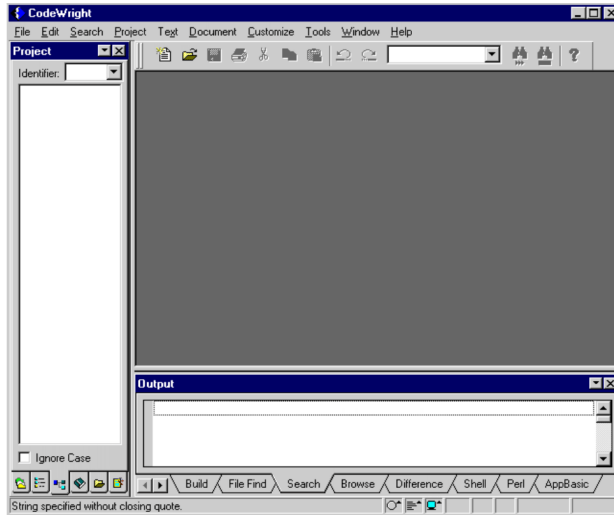
For more information about CodeWright's Sync technology, see the *User's Guide* chapter on *Synchronization*.

- **The CodeSense Libraries Wizard:** Accesses the **CodeSense Global Configuration** dialog where libraries that point to designated C/C++ and Java source files are created. The libraries are used by CodeWright's CodeSense feature. See the *User's Guide* chapter on *Editing & Printing* for information on CodeSense.

A First Look

After completing any necessary configuration tasks, and accessing needed help tips, the **Configuration Wizard** and **Tip of the Day** dialogs can be closed. The resulting CodeWright screen displays a menu bar at the top, with the **Standard** toolbar docked directly beneath. The window docked on the left is the Project Window, and the window docked on the bottom is the Output Window. All of these items will be described briefly in the remainder of this manual.

Initial CodeWright Screen



Two files open by default with the initial CodeWright session: README.TXT and UPDATE.TXT. README.TXT contains important information about the latest release, and UPDATE.TXT lists the latest features. It's a good idea to read through these documents before closing them.

The Menu Bar

The menu bar consists of ten items, described in the following topics.

File Menu

The first menu is the **File** menu. The **File** menu is intended for standard file operations, i.e. **New**, **Open**, **Save**, **Print** etc. Additional items are described (or deferred to the *User's Guide* for more information) below:

- **Difference**, and **Merge**. The **Difference** and **Merge** utilities are described in detail in the *User's Guide* chapter on *Checking and Reformatting Files*.
- The **Find** menu-item is used for finding files on any storage medium. Standard DOS wildcard characters can be used when searching for files with this option.
- **Send Mail** sends files via MS mail or cc: Mail, and **Reload** reloads the current file from disk.
- **Change Directory** accesses a dialog that allows the current directory to be changed.
- The **Filters** dialog defines file specifications or wildcard patterns, such as *.c, to filter files of interest from otherwise long lists of files. It's a good idea to set the filters during the initial CodeWright session so that they'll be ready to use in the future. Here are a few useful filters:

Description	Pattern
Text files	*.TXT;*.RTF
Initialization files	*.INI;*.CFG
Make files	*.MAK;*.

- The last two items on the **File** menu, **Print** and **Print Setup**, are described in more detail in the *User's Guide* chapter entitled *Editing & Printing*.

Note: An optional **FTP** item can be added to the **File** menu by loading a library in CodeWright's **Customize|Libraries** dialog. The **FTP** item can be used for transferring files from CodeWright to remote hosts. For more information on CodeWright's FTP feature, refer to the *User's Guide* chapter on *File Loading, Backup and FTP*.

Edit Menu

The first two sections of the **Edit** menu contain standard editing items, such as **Undo**, **Redo**, **Cut**, **Copy** and **Paste**. Three items that may not be familiar are the **Scrap Buffer**, **Append**, and **Erase** items:

- The **Scrap Buffer** item selects the **Scrap Buffer** for cut/paste operations. The **Edit** menu also has a **Clipboard** option, allowing you to choose which copy/paste repository you want to use. CodeWright allows multiple Clip and Scrap buffers to be used, the contents of which can be viewed in a **ClipView** Window. The **ClipView** Window is an optional tab on the Output Window (described later), which is turned on in the **Customize | Libraries** dialog.
- The **Append** item adds selected text to the end of the selected clipboard/scrap buffer.
- The **Erase** item clears the selected clipboard/scrap buffer. CodeWright's Scrap and Clipboard features are both described in more detail in the *User's Guide* chapter on *Editing & Printing*.

The third section of the **Edit** menu has options for inserting items into the current buffer.

- **Insert File** brings up a dialog for selecting a file to be inserted into the current document.
- **Insert Literal** brings up a dialog for entering characters into the document that would otherwise trigger a command. The literal may be chosen using ASCII, Hex, or Decimal values.
- **Insert Link** inserts a button link at the cursor position in the current document. **Button links** are special action buttons that CodeWright lets you embed in your text files. You may use them to view bitmapped images, bring up a related document or spreadsheet, run a macro or just to make notes. **View Links** (the last item on the **Edit** menu) is used for viewing the contents of the **Button Links** database.

Button Links are described in more detail in the *User's Guide* chapter on *Search and Replace and Navigational Tools*.

In the Macros section of the **Edit** menu, use **Record**, **Playback**, **Keystroke Macros**, and **Run Key Macro** to make, test, edit and use Keystroke Macros. CodeWright's Keystroke Macros are described in the *User's Guide* chapter *Custom Interface*.

Search Menu

The **Search** menu contains all the items necessary for performing search and replacement operations on single or multiple files. All of CodeWright's search features are described in detail in the *User's Guide* chapter on *Search and Replace and Navigational Tools*. The one item worth describing here is **Options. Search|Options** brings up the **Default Search and Replace Settings** dialog, where you set up searches to operate the way you expect. It is prudent to set search options before editing so that some of the following issues might be addressed:

- Should the search be case sensitive?
- Should there be a prompt on replacement by default?
- Should regular expression searching be enabled?
- Should matching text be selected (highlighted)? If so, should it be momentary or continuous? If continuous, should **Restrict to Selection** be turned off so that **Search Again** will work as expected?
- Should the word that the cursor sits on be in the **Search** dialog by default, or should the word be typed?

Project Menu

The **Project** menu is used for creating, opening, closing and manipulating projects, project spaces, and workspaces:

- Projects are used to organize and store individual sets of files and configuration settings. A project, at a minimum, is a list of files that you find it useful to group together logically.
- Project Spaces organize sets of projects. Each project must be part of a project space. If an existing project is opened without first creating a project space, a project space will automatically be created to envelop the project.
- Workspaces preserve options and settings for the currently open windows and documents; this can include project and non-project files. Reload the workspace to pick up where you left off with those files.

Options on the **Project** menu include:

- **Set Current**, for setting a project to be current in the present project space.
- **Project Space**, for accessing a submenu to create, open and close project spaces and to create, add, and remove projects for the current project space.
- **Compile, Compile (Debug), Build, Rebuild, Rebuild (Debug), Debug, and Execute**, all of which have to be configured in the **Project|Properties** dialog in order to use them. More information about CodeWright projects are provided in the *User's Guide* chapters on *Build your First Project* and *Projects, Project Spaces, and Workspaces*.

Text Menu

The **Text** menu has various items for formatting and editing text. Some of these items, like **Selective Display**, the **Comment** items, and **Format Source**, are discussed in-depth in various *User's Guide* chapters. The other items are fairly self-explanatory and will be briefly described here.

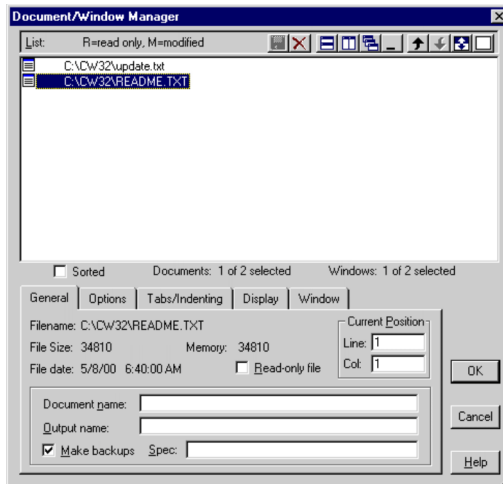
- **Word Wrap** enables and disables automatic line wrapping. The lines will wrap at the right-margin mark, which is set on the **General** tab of the **Customize|View Setups** dialog.
- The **Upper** and **Lower** menu items will respectively capitalize or un-capitalize selected text.
- The **Slide In** and **Slide Out** menu items will slide the current line or the current selection right (in) or left (out) the length of a tab stop. The **Prompted Slide In/Out** items prompt the user for some text that will be used to slide the text in or out.
- The **Left Justify**, **Right Justify**, and **Center** items will position text within a column or line selection as their names describe.
- The **Enumerate** menu item inserts line numbers in ascending or descending order for selected text, and **Format Columns** aligns columns within a selected block.

Document Menu

The **Document** menu has items that access dialogs and perform operations that are commonly used on documents. The first and second sections contain fairly self-explanatory items for creating, closing, clearing and maneuvering documents. The third section has items for inserting and moving between bookmarks. Bookmarks are described in detail in the *User's Guide* chapter on *Search and Replace and Navigational Tools*.

The last item on the **Document** menu accesses the **Document/Window Manager** dialog. The **Document/Window Manager** dialog lists documents and windows that are currently open in CodeWright. It can be used to change the attributes of the open windows and documents on an individual basis. It contains a window listing the files and windows currently open, within which files can be selected for manipulation. The window's file-icons represent the different document/window-states that the files are in (as described in the online help topic *Document/Window Manager Dialog*), as well as their read-only status.

Document/Window Manager




Some of the things that can be done in **Document/Window Manager** include:

- Specifying indent options.
- Tiling/cascading individually selected documents or windows.
- Setting different **View Setups** (fonts, colors, etc) for individual documents/windows.
- Changing display modes (e.g. **Selective** or **Hex** displays).
- Changing window attributes (e.g. scrollbars, line numbers).
- Setting soft line **Wrap** (located on the **Display** tab) - wraps lines at a specified column without inserting a line feed.

All operations performed from **Document/Window Manager** can be performed on a per-document or per-window basis.

Customize Menu

The items on the **Customize** menu access dialogs used for configuring various parts of CodeWright.

Note: Right-click  on the CodeWright status bar to quickly access many of the same items that are available on the **Customize** menu.

Items on the **Customize** menu are as follows:

- The **Environment** item accesses the **Environment** dialog, which has a number of tabs for manipulating the CodeWright environment. For example, you can set backup files and backup specifications on the **Backup** tab of the **Environment** dialog, and you can modify menus on the **Menu** tab.
 - ✓ Portions of the **Environment** dialog are discussed in the *User's Guide* chapter on *Command Key, Libraries, & Environment*. However, since the tabs in the **Environment** dialog relate to a wide variety of features in CodeWright, the items in the dialog will be covered extensively throughout the *User's Guide*.

The **General** tab of the **Environment** dialog is used for setting various items in the CodeWright environment. Concerns addressed by the settings in this dialog include the following:

- ✓ Whether system prompts (e.g. **API Command** dialog/prompt) should pop-up in a dialog, or be displayed on the status line. Often, the preference is the status line prompt, the setting for which is **Use Command Line Prompt**.
- ✓ Whether windows and documents operate as a single unit, as they do in most Windows applications, or independently. The toggle for this functionality is **One Document Per Window**.
- ✓ Whether a list of recently loaded files should appear at the bottom of the **File** menu. This option can be quite handy for reloading files, and is turned on by marking **Show File List on File Menu**.
- ✓ Various file loading and file validation concerns.
- ✓ Whether the **Vertical/Horizontal Tile** options on the **Window** menu should truly tile vertically or horizontally, or whether they should tile in the traditional Windows way.

The **State** tab allows you to save state information between CodeWright sessions. Ask yourself a couple of questions:

- ✓ Do you want CodeWright to remember and load the last file you were working on, all files, or no files?
- ✓ Do you want to save bookmarks between sessions?
- The **Read Configuration Data...** item on the **Customize** menu accesses the **Read Configuration Data from a File** dialog. The dialog can be used to read CodeWright configuration settings from any file that contains them.
- The **Libraries** item is used for interactively loading CodeWright DLLs (or Add-Ons) to extend the functionality of the CodeWright program. A number of commonly used Add-Ons are contained in the list of **CodeWright Libraries** on the **Load CodeWright DLL** dialog. **CodeWright Libraries** can be quickly and conveniently loaded by placing checkmarks next to the ones to be loaded. More Add-Ons are available on Starbase's WebPages, at <http://www.starbase.com>.

- The **Toolbars** and **Keyboard** menu items bring up the **Toolbar Customization** and **Assign Keys** dialogs, respectively. Descriptions for how these dialogs are used can be found in the *User's Guide* chapter on *Custom Interface*.
- The **View Setups** option accesses a dialog used for viewing and changing view setups, which store sets of document and window preferences, such as colors, fonts, line numbers, and scroll bars.
- The **ChromaCoding Lexers** option accesses the **ChromaCoding Lexer Settings** dialog. Here you can create or modify lexers to color the various elements of your programming language "vocabulary". The vocabulary of a language includes identifiers, braces, preprocessors, keywords, operators, strings and comments.
- The **Language** option accesses a dialog that is used for controlling language-specific features, such as ChromaCoding (syntax coloring), and template expansion.

Note: **View Setups**, **ChromaCoding Lexers**, and **Language** are described in the *User's Guide* chapter on *View Setups and Language Support*.

- The **CodeSense Global Configuration** option accesses the **CodeSense Global Configuration** screen. When properly configured, CodeWright's CodeSense feature completes the names of, and suggests parameters and members for, code elements as they are being typed. It is available for C/C++ and Java files. CodeSense works by parsing the programming libraries that you add on the **CodeSense Global Configuration** screen, as well as project source files and files that are open in CodeWright. CodeSense must also be configured on the **Customize|Language|CodeSense** dialog to be used. For more information on CodeSense, refer to the *User's Guide* chapter on *Editing & Printing*.
- An additional **CobolExt Settings** item is available on the **Customize** menu when **COBOL Extensions** is loaded in **Customize|Language**. See the topic *COBOL Extensions* in the *User's Guide* chapter *Editing and Printing*.

Tools Menu

The **Tools** menu has a variety of tools for manipulating and extending CodeWright. Items on this menu are described/referenced below:

- The first five items on the **Tools** menu, (with the exception of **API Command**), are for CodeWright's macro languages:
 - ✓ The **API Macros** and **Run API Macro** menu items are used for making and running CodeWright API Macros.
 - ✓ The **AppBasic Macros** and **Perl Macros** menu items both have submenus containing items for loading and working with AppBasic and Perl Macros.

Macros are described in greater detail in the *User's Guide* chapter entitled *Extend CodeWright*. (The **API Command** option is described in the *User's Guide* chapter on *Command Key, Libraries & Environment*.)

- The **Version Control** item on the **Tools** menu accesses a submenu containing items that either run version control operations or that access dialogs used for running or setting up version control in CodeWright.

When looking at this submenu, it is important to remember that CodeWright does not come with its own version control utility. Instead, it integrates with existing version control systems using one of two methods: command line integration or SCC API integration. More information about using version control from within CodeWright can be found in the *User's Guide* chapter on *Version Control*.

- The **Spell Check** option on the **Tools** menu accesses CodeWright's spell checker. CodeWright's spell checker is described in more detail in the *User's Guide* chapter on *Checking and Reformatting Files*.
- The next three items are handy utilities for performing sundry tasks:
 - ✓ The **Filter** option brings up a dialog that lets you perform an external operation on the current document or a selected portion of it.
 - ✓ The **Shell** option runs a command shell.
 - ✓ The **Shell Command** option brings up a dialog that invokes a Windows/DOS application.
- The last section of the **Tools** menu is optional. It contains tools that may vary if CodeWright has been customized. Custom tools are set up in the **Project|Properties** dialog on the **Tools** tab. These are described more comprehensively in the *User's Guide* chapter on *Projects, Project Spaces, and Workspaces*.

Window Menu

Similar to the **Document** menu, the **Window** menu is used for manipulating and creating windows opened in CodeWright. It is important to understand how CodeWright differentiates between "windows" and "documents". Depending on the configurations that have been made in CodeWright, the files that are open may not always be contained in their own windows. The difference between windows and documents is described below.

The Difference between Windows and Documents

Some confusion may arise when attempting to discern how CodeWright differentiates between "windows" and "documents". Depending on the keymap being used, documents in CodeWright may not all appear in their own window.

- Most of the keymap emulations that CodeWright provides (e.g. BRIEF, Epsilon, and vi) use one window for all documents by default. While there may be 10 documents open, only one of them will be seen at a time in the one window that is open. This characteristic defers to the DOS days of the editors from which the referenced keymap emulations were derived.
- The CUA keymap, on the other hand, uses one window for every document. Therefore, in CUA, multiple windows will be seen, each containing its own document.

The "document per window..." characteristic can be toggled for any keymap using the **One Document per Window** option on the **Customize | Environment | General** dialog. Keymaps are described in more detail in the *User's Guide* chapter on *Custom Interface*.

Details of the Window Menu

The items on the **Window** menu are fairly self-explanatory:

- **New Window** creates a new window. If a document is open, the new window will contain a second instance of that document.
- **Full Screen** puts documents into full-screen mode, eliminating clutter caused by menus and toolbars, and maximizing the space available for editing. To exit full screen mode, click on the text 'Restore Desktop' displayed in the status bar.
- The **Tile** and **Cascade** items are alternate ways to arrange windows on the screen.
- **Arrange Icons** arranges minimized windows at the bottom of the CodeWright screen. The **Close All** option closes all windows that are open.
- The **Project** and **Output** options toggle the Project and Output Windows off and on. A brief description of these windows is provided in the topic *Project and Output Windows*, in this chapter.
- The **Manager** option accesses the **Document/Window Manager** dialog. This is the same dialog that is accessed when clicking the **Manager** option on the **Document** menu. For more information about the **Document/Window Manager**, see the topic *Document Menu*, in this manual.

Help Menu

Use CodeWright's **Help** menu to access and configure CodeWright's online help and API Assistant. More information on Help and the API Assistant can be found in the *User's Guide* chapter on *Editing & Printing*.

Output and Project Windows

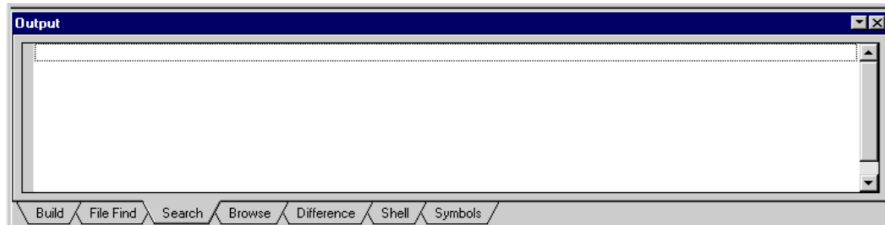
The default CodeWright screen has two system windows, one docked on the left, and one on the bottom. The window docked on the bottom is called the Output Window. The window docked on the left is called the Project Window. These two windows can be "undocked" by dragging them away from any edge of the CodeWright screen.

- Further information on customizing the Project and Output Windows can be found in the *User's Guide* chapter on *Custom Interface*.
- The Project Window is described in greater detail in the *User's Guide* chapter on *Projects, Project Spaces, and Workspaces*.
- The Output Window is described to a greater extent in various chapters of the *User's Guide*, as it contains a number of tabs that relate to various features in CodeWright.

Output Window

The Output Window initially has seven tabs and is docked on the bottom edge of the CodeWright screen. Three more tabs can optionally be added by loading the Add-Ons for the **AppBasic** and **Perl** macro languages, and the Add-On for the **Clipboard/Scrap Viewer**. Add-Ons are loaded using the **Customize | Libraries** dialog.

The Output Window



Uses for the tabs of the Output Window vary, but they are generally used for displaying and manipulating the results of operations in CodeWright.

The tabs of the Output Window are:

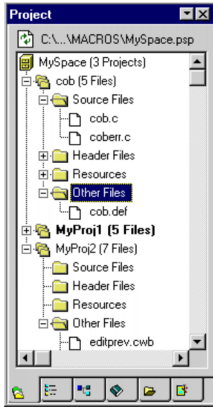
- **Build** tab: Displays the results of builds or compiles.
- **File Find** tab: Displays the results of file searches produced by **File | Find**.
- **Search** tab: Displays the results of string searches produced from search operations.
- **Browse** tab: Displays tags or browser database files.
- **Difference** tab: Displays the differences of two files, side-by-side.

- **Shell** tab: Acts as a virtual DOS shell.
- **Symbols** tab: Allows you to navigate through files, much like the **Browse** tab.

The Project Window

The Project Window is initially docked on the left edge of the CodeWright screen.

Project Window



Like the Output Window, the Project Window contains several tabs. The tabs are for viewing information about projects and other files. The tabs (from left to right) are:

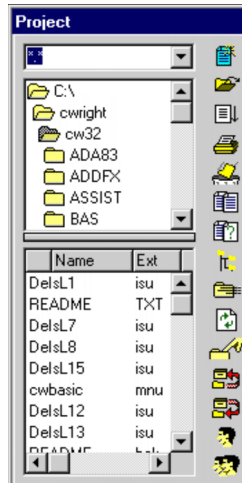
- **File View** tab: Lists the projects in the current project space with their member files in hierarchical form underneath. It lets you operate on individual files or groups of files within current projects. More information about the **File View** tab can be found in the *User's Guide* chapter on *Projects, Project Spaces, and Workspaces*.
- **Outline** tab: Presents a hierarchical view of Symbols in project files and any other files that are currently loaded. More information about Symbols and the **Outline** tab can be found in the *User's Guide* chapter on *Search and Replace and Navigational Tools*.
- **Objects** tab: The **Objects** tab displays a hierarchical view of C/C++ and Java objects/symbols. The objects/symbols displayed are associated with the name that is typed in the **Identifier** box at the top of the window. The window can be used to view and browse code. More information about the **Objects** tab can be found in the *User's Guide* chapter on *Search and Replace and Navigational Tools*.
- **Bookmarks** tab: Gives a view of local and global bookmarks defined in documents. More information about Bookmarks and the **Bookmarks** tab can be found in the *User's Guide* chapter on *Search and Replace and Navigational Tools*.

- **Open** tab: Acts like a persistent File Open dialog and file manager in one. It presents a list of icons representing valid drives from which to choose, a directory tree, a place to specify a file filter to limit the files displayed, and a box where matching files are listed. Double-click on any of the files listed to load it for viewing or editing. More information on the **Open** tab is provided in the next section: *More on the Open tab*.
- **CodeFolio** tab: Presents a directory tree of pre-defined chunks of text that can be specific to programming languages. The text can be inserted into your current document. Custom and user-defined Snippets can also be added to the window. Refer to the topic *CodeFolio Snippets* in the *User's Guide* chapter on *Editing & Printing* for more information.

More on the Open Tab

The Project Window's **Open** tab is a handy tool for file-operations. Opening files is just the beginning of what it can do. There are a total of 20 other operations that can be performed, as represented by a series of icons on the tab's frame. Tool tips (small popup messages) describe each of the icons.

Project Window: Open Tab

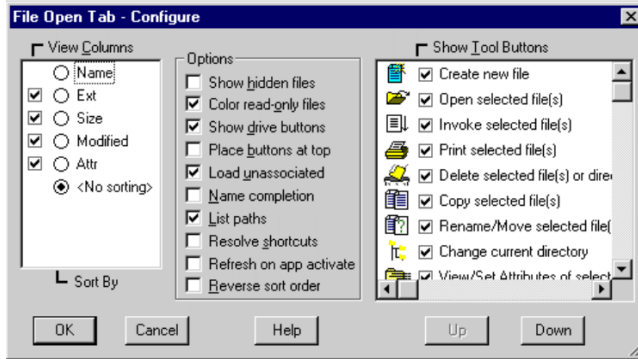


Use the **File Open Tab - Configure** dialog to remove, add, or reconfigure the icons on the **Open** tab. To access this dialog, click on the **Configuration Dialog** button:




The following displays:

File Open Tab – Configure



Two of the icons you can set for the **Open** tab are for user-defined commands: 

and . User-defined commands can be set to perform any operation on files selected within the window. The user commands must be CodeWright API function calls, rather than DOS commands or the like. You may, however, execute DOS commands by specifying the API function `ExecUserCmnd` (help on `ExecUserCmnd` or any other CodeWright API can be found in CodeWright's online help).

Example: `ExecUserCmnd "Dir"`

The **File Open Tab - Configure** dialog is also used to set whether or not to display hidden files, file timestamps, file sizes and file attributes. The timestamps, file sizes and file attributes are only visible as Tooltips that display when the mouse cursor pauses over a file in the window.

The Standard Toolbar

CodeWright has several toolbars, but only the **Standard** toolbar is turned on by default. Since the **Standard** toolbar is part of the default CodeWright screen, it will be introduced here; all of the toolbars are described in more detail in the *User's Guide* chapter on *Custom Interface*. A description of the **Standard** toolbar follows.

The Standard Toolbar



The **Standard** toolbar consists of the following:


- The first two buttons on the **Standard** toolbar are forward and back buttons. They allow documents and document positions to be navigated based on a history of documents and positions that have been accessed during the current CodeWright session. Drop-down lists are available for each button. The drop-down lists allow specific documents to be accessed without having to cycle through all the documents in the list.
- The nine buttons that follow the forward and back buttons are for editing and file operations: creating, opening and saving files; printing; cutting, copying and pasting text; and redo/undo of the last change made.
- The next item on the **Standard** toolbar is the **Toolbar Search Box**. The **Toolbar Search Box** refers to the search capability built into the **Standard** toolbar in the form of a drop-down list box control. It provides the most immediate, convenient way to perform simple searches. The **Toolbar Search** honors all of the settings in the **Search | Options (Default Search and Replace Settings)** dialog.

To use the **Toolbar Search**, type in the string you wish to search for, and press




- The next two buttons on the **Standard** toolbar are also related to searching:



- ✓ Press  to repeat the last search



- ✓ Press  to search for the word under the cursor in the current document.

- The last button on the **Standard** toolbar (the question mark) is for accessing CodeWright's online **Help**.

Index

A

- Add-Ons/DLLs/Libraries
 - Additional Add Ons on the CD 24
 - Loading 24
- Answer Wizard 17
- Autorun 7

B

- back to previous document position 32
- BRIEF command set 10
- Buffers Limit 2
- Button Links
 - Insert Link 20
- Buttons 20
 - action 20

C

- Case
 - Changing Upper, Lower 22
- CD-ROM version 7
- Centering text 22
- ChromaCoding Lexers 25
- Class browser 29
- Clipboard/Scrap Buffer Size 2
- CodeFolio Tab
 - Snippets Directory 30
- CodeSense 25
- CodeWright Libraries 24
- Columns
 - Format Columns 22
- Command line
 - options 14
- Comments
 - inserting 22
- Compiling 12
- Configuration
 - Files 7, 12

- Wizards 17
- CUA command set 10
- Custom installation 9
- Custom tools 26
- Customization Button
 - CodeWright Status Bar 23
- Customize Menu 23
- CWINI 12
- CWLEX 12
- CWLIB 13
- CWMENU 13
- CWPST 13
- CWRIGHT.I60 7
- CWRIGHT.INI 7, 14
 - location 6
- CWRIGHT.MNU 14
- CWRIGHT.PST 7, 14
- CWSENSEINI 13
- CWSENSELIBDIR 14
- CWTMP 12

D

- DLLs/Libraries/ Add-Ons
 - loading 24
- Document
 - Difference between windows and documents 26
 - move forward and back 32
- Document Menu 22
- Document/Window Manager Dialog 23

E

- Edit Menu 20
- Electronic Mail
 - Sending Email to Starbase 3
- Email 3
- Emulation
 - select CUA, BRIEF, vi, or Epsilon 10
- Environment variables 5, 12
- Environments
 - Supported for Synchronization 10
- Epsilon command set 10

F

- Fax Number 3
- File
 - finding 19
- File Menu 19
- Files
 - Attributes - Open Tab 31
 - File Manager 30
 - File Open dialog functionality 30
 - Limit on Number of Files 2
 - Open tab -- Project Window 31
 - private 5
 - Reloading current file from disk 19
 - Sending via Email 19
 - Show File List on File Menu 24
 - Size Allowed 2
- Filter 19, 26
- Filters
 - File|Filters dialog 19
- Finding Files
 - File Find tab 28
 - File|Find 19
- Format Source 22
- forward to next document position 32
- FTEE Utility 12
- Full Screen
 - Window menu 27

H

- Help
 - Standard toolbar 32
- Help Index File Wizard 17
- Help Menu 27
- Hidden files
 - Open tab-- Project Window 31
- Home directory 14

I

- Indenting
 - Prompted Slide In/Out 22
 - Slide In 22
 - Slide Out 22

- Insert
 - Button Links 20
 - File 20
 - Literal (Hex, Decimal, ASCII) 20
- Insert line numbers
 - Enumerate 22
- Installation
 - AUTOEXEC.BAT 5
 - AutoMerge 6
 - CodeWright Program Group 10
 - CONFIG.SYS 5
 - Considerations 11
 - Directory 8
 - disk space 9
 - installation directory 8
 - Installing Synchronization 10
 - Options (Compact, Custom, Full, Typical) 9
 - serial number 7
- Internet Mail 3

K

- Keyboard
 - Assign Keys 25
- Keymap
 - selecting initial 10
- Keystroke Macros 20

L

- Language
 - Tools|Customize Menu 25
- Left Justify 22
- Libraries/ Add-Ons/DLLs
 - loading 24
- Limits 2
- Line
 - Numbering 22
- Lines
 - Length Limit 2
 - Limit per Buffer 2
- Link
 - Insert Button Link 20
- Literal
 - Insert Hex, Decimal, or ASCII Liter-

al 20
Long filenames 8

M

Macros
 Tools Menu options 25
Mail
 Sending Email to Starbase 3
 Sending files 19
Memory requirements 1
Menu Definition file 14
Menus
 Customize 23
 Document Menu 22
 Edit Menu 20
 File Menu 19
 Help 27
 Project Menu 21
 Search Menu 21
 Text Menu 22
 Tools Menu 25
Merging Changed Source Files 6
Mode, Windows 1
Multiple Projects and Configurations
 14

N

Navigate
 move between documents 32
Networks
 Installing on 5, 15
Number of Files 2

O

One Document per Window 27
Open tab
 Project Window 30
Output
 splitting 12
Output Window 28
 Browse tab 28
 Build tab 28
 Difference tab 28

Index

File Find tab 28
Search tab 28
Shell tab 29
Symbols tab 29

P

Phone Number 3
Popup Menu Definition file 13
PREVIOUS subdirectory 7
Previous versions of CodeWright 5
Private files 5
Project and Output Windows 28
 toggle 27
Project Menu 21
Project Window 29
 Bookmarks tab 29
 CodeFolio tab 30
 File View tab 29
 Objects tab 29
 Open tab 30
 Outline tab 29
Projects 14
Prompts
 System 24

R

README.TXT 3, 11, 18
Reload 19
Right Justify 22
Running Setup 7

S

Scrap Buffer
 Size allowed 2
Search Menu 21
Search Options 32
 Case Sensitivity 21
 Dialog 21
 Prompt On Replacement 21
Search String Length 2
Selecting Initial Keymap Command
 Set 10
Selective Display 22

- Send Mail 19
- Serial number 7
- Setup 5
- Shell
 - Tools Menu option 26
- Shell Command
 - Tools Menu option 26
- Snippets
 - Directory on the CodeFolio tab of the Project Window 30
- Source Files
 - merging changed 6
 - user modified 6
- Spell Check
 - Tools Menu option 26
- Standard Toolbar 31
 - Help button 32
 - repeat last search button 32
 - search for word under cursor button 32
- State File 13, 15
- Status Bar
 - Customization Features Shortcut Button 23
 - Restore Desktop from full screen mode 27
- Synchronization
 - Supported Environments 10
 - Wizard 17
- System Requirements 1

T

- Technical Support 3
- TEMP 12
- Temporary Files 1, 12
- Text Menu 22
- through 32
- Timestamps
 - Open tab-- Project Window 31
- TMP 12
- Toolbar
 - forward and back button 32
- Toolbar Search Box
 - Standard Toolbar 32
- Toolbars
 - Customization 25

- Standard toolbar 31
 - undock 28
- Tools
 - Custom 26
- Tools Menu 25

U

- Undo 32
- UPDATE.TXT 11, 18
- Upgrading From Previous Version 5
 - UPDATINI.EXE 5
- User-defined commands
 - Open tab--Project Window 31
- Utilities
 - Tools Menu options 26

V

- VDOS 12
- Version Control
 - Tools Menu option 26
- Versions (previous) 5
- vi command set 10
- View Setups 25

W

- Web Page 3
- WildCards
 - File Filters 19
 - Patterns 19
- Window Menu 26
 - Arrange Icons 27
- Windows
 - Difference between windows and documents 26
 - Limit 2
 - mode required 1
 - move forward and back 32
- Windows 9x 7
- Windows and Documents
 - One Document Per window 24
- Wizards
 - configuration 17
- Word Wrap 22