Table of Contents

Introduction

CRiSP has been designed as a platform independent, general purpose file editor for plain text or data files - these can be larger than 100 Mbytes in size.

CRiSP provides a large number of intuitive and easily accessible features making editing of files a more efficient process - thus making you more productive.

CRiSP is available on a wide range of hardware platforms and operating systems. All versions are very similar in operation and behaviour. This guide is aimed as an introduction to the most commonly used features of CRiSP and is a useful starting point if you are new to the product.

CRiSP is a feature-rich and sophisticated product with many facilities. The best way to learn the product is to learn it a little bit at a time, much as you would do for any other application. You can use the on-line help to find out details about some of the more rarely used operations.

You can start using CRiSP straightaway - the most basic facilities are easily useable with no prior knowledge or experience of CRiSP. It is only when you come to do more repetitive tasks that you may need to delve into the help information available.

Configurability

CRiSP provides a variety of functions and options, many of which are designed to give you, the user, the flexibility and choice in how you work. You are not constrained to specific implementation choices, but instead can configure CRiSP to look and feel exactly how you want.

There are many of configuration options which allow you to configure CRiSP to best suit your way of working. Please refer to the User Guide, available in the on-line help and from the **Help** button of the various dialogue boxes, in order to explore these configuration options.

CRiSP often provides up to three ways to achieve the same thing. The most common commands area available from the menu bar, a toolbar icon, or from the function key. You decide which is best for you.

Platform Specific notes

Most of the information in this guide is common across platforms. Occasionally there will be platform specific issues, which will be pointed out. Generally speaking all UNIX versions operate in a similar way, and by contrast, the Windows/95 and Windows/NT versions are slightly different.

Where needed, the term **Microsoft Windows** is used to mean any of the common Microsoft operating system environments.

Further Information

This guide does not attempt to be exhaustive in the information it provides. CRiSP provides a complete set of on-line documentation which can be accessed from the **Help** menu. You are advised to visit the help menu and become acquainted with the facilities available in CRiSP.

Character mode and GUI Mode

CRiSP exists as two distinct programs, one of which is a graphical program; the other is a character mode program, e.g. suitable for use inside an xterm window or a command window. This document only describes the GUI mode version of CRiSP.

The character mode version is limited in its graphical interface by virtue of the fact that it runs solely inside a **COMMAND.COM** or **xterm** window, i.e. there is no scrollbar, toolbar or menu bar support.

You are referred to the on-line User Guide for more details about the character mode version.

Character mode popup menus

You can configure CRiSP so that some function keys pop up dialog boxes, whilst others use a more terse command line prompt. The trade off is simply that one involves more mouse co-ordination, whereas in character mode, it is easier to accomplish things without taking your hands off the keyboard.

For some functions, CRiSP makes use of character mode popup menus rather than the usual GUI popups. For example, pressing **<Tab>** at the **Edit file** prompt (accessed via **<Alt-E>**) pops up a window showing a list of files.

The **<Ctrl-A>** key can access what is known as a delayed menu. A delayed menu is a menu that does pops up after a certain delay. If you type quick enough, then the menu is not popped up, thus speeding the edit operation. But if you are uncertain of an option then it is useful to have CRiSP prompt you with the valid options available.

You can configure CRiSP to use these character mode popup functions even if you are using the GUI version of CRiSP.

Invoking CRiSP

Depending on the platform you use, there are different ways to invoke CRiSP. Generally speaking there are two main ways:

1. clicking on an icon on the desktop
2. typing in at a command terminal window

Microsoft Windows:

When installed onto a Windows/95 or Windows/NT machine, clicking on the CRiSP icon is the easiest way to start CRiSP, since the default installation will place an icon on your desktop.

Under Windows/95, the program to invoke is called **CRiSP.EXE.** You will need to ensure that the CRiSP distribution is in your PATH if you are going to invoke CRiSP from the command window. The CRiSP binary is located in the **BIN.W32\** directory.

UNIX:

Under UNIX, unless you have a desktop manager, or have configured your workstation, then the most common way to invoke CRiSP is from a command window (normally an **xterm** window). The program to invoke is called **crisp** and is located in the directory **bin.???/** directory of the installation. The name of the directory is platform specific, and allows for multiple versions of CRiSP to be installed on a file server, serving different types of workstations.

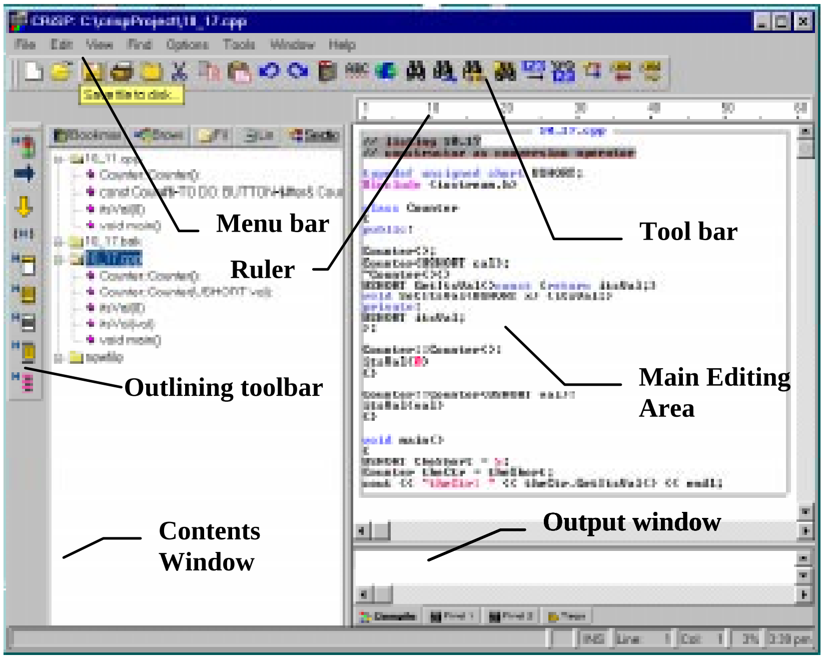
If you are invoking CRiSP from a command line prompt, then you can also specify one or more files on the command line. If you do not specify any file names on the command line, then CRiSP will automatically load the last file edited.

Under UNIX, you can specify wild cards to force CRiSP to load the files specified.

Under Microsoft Windows, you will not be able to use wild cards to load files, and may need to enclose any filenames specified on the command line in double quotes, if, for example, the file to be loaded contains a space in the filename.

Screen components

The following picture illustrates a typical screen-shot of CRiSP. This screen shot was taken on a Windows/95 machine, and although there will be some small differences for UNIX, these differences are minor and do not really affect the documentation.



The main parts of the window are designed to be - where possible, CRiSP respects the conventions used for user interface design. For example, you will find a familiar set of operations hanging off the file menu.

If you have not used PC applications before and find the interface unfamiliar, then do not worry. Everything you do can be undone and you are given ample opportunity to cancel any operation you initiate. You are encouraged to explore and experiment .

Menu bar The menu bar allows you to access most of the common and frequently used commands available in CRiSP, such as opening a file, displaying help, performing a compilation, etc. The menu bar is divided into a number of different categories, the majority of which will be familiar to you if you have used any other PC type applications.

You are encouraged to browse the menu items and learn the rough position of the commands and features available.

The menu bar is useful for accessing less frequently used facilities. Commands which are likely to be used frequently (such as deleting a word) are assigned special keystrokes to make it faster for proficient typists to use.

Tool bar The tool bar contains a selection of icons which perform tasks similar to those available on the menu bar. The tool bar commands are designed for those more frequently used commands. Since the icons take up much more space than the menu bar, which has nested menus, there are much fewer icons available in the default configuration. You can configure the commands available from the **Options->Toolbar** menu.

The toolbar icons provide a *tool-tips* mechanism which appear when the mouse hovers over an icon for more than about 1/4 of a second. These little popup messages are reminders of the icons function.

Outlining toolbar This allows you to hide and reveal portions of your file or program according to various criteria. See also View>Outline menu items for other choices.

Ruler The ruler is an occasionally useful facility in CRiSP. It allows you to keep track of the exact column position of text in the edit window. If you click and hold down the left mouse button on the ruler, a grid line is drawn which is useful for ensuring that text is aligned properly, especially if you are using a 17" or larger monitor and a tall window.

Scrollbars The scrollbars are standard mechanisms that allow you to use the mouse to *pan* through the current file being edited.

Output Window This is used to show results of Compilations, complex Searches through more than one file or buffer (initiated from the Find>Search files or > Search buffers menu). Or to show output from popping tags in the Tags file.

Contents Window This is resizeable and can display five different elements of CRiSP.

a) Bookmarks. These can be within a file, within a defined Project or between any otherwise unrelated files.
b) Browser. First define a ‘tags’ file (see online help) then use this Browser to jump to any type of Object or structure within the file, or a defined project or the whole file system.
c) File System Navigator. Allows you to quickly navigate the file system. Double clicking loads the file into a new Buffer
d) Links. A link can be dropped as a button into a file and used simply as a reminder message or could be used to execute another program or macro.
e) Sections. Used to view the functions, etc of a program file. Double clicking takes you to that point in the current buffer.

Status bar The status bar provides various useful status information about the current file being editing, such as line and column number information

The Tool bar

The toolbar is a useful collection of functions which are used frequently. Many of the items on the toolbar duplicate entries available in the menu bar but are quicker to access, and are more standard across applications, especially the open file, save file, and cut/copy/paste functions.

The toolbar is configurable (see the **Options->Toolbar** dialogue box). You can configure the size of the icons (large or small), select from different collections of icons, or reassign functions to the icons.

Most users either keep the default toolbar or hide it from view because they never use it.

The following table summarises the standard icons on the toolbar. You do not need to remember the functions of each icon, since CRiSP supports floating tool-tips which appear if you let the mouse hover over an icon for more than a short period of time.

File Manipulation

Create a new buffer.

Open a file for editing.

Save modified file to disk.

Print file.

Editing

Copy selected area to the scrap buffer.

Cut selected area to the scrap buffer.

Paste contents of the scrap into the current buffer.

Undo last change to the current buffer.

Redo the last undo.

Searching & Replacing text

Search for a string. Selecting this brings up the Search for a String dialogue box.

Repeat last search command. Selecting this brings up the Search & Replace dialogue box.

Search and replace text strings.

Search loaded buffers

Line numbers

Go to a specific line/column number.

View line numbers,

Show routines/sections in the buffer.

Tags

Jump back to previous tag.

Jump to function under cursor.

Miscellaneous

Scroll lock the current buffer.

The Menu bar

The menu bar appears at the top of the CRiSP editing window and provides access to a large number of functions and dialogue boxes. Some of these are common to any other GUI application, and are designed to be instantly recognisable, following standards where they apply. Other functions are specific to the CRiSP application.

Select an entry from the menu bar image or one of the sections below for details on the options available.

File This menu is used to perform file oriented actions amongst other things. The following is a summary of the items available on the menu bar.

New Creates a new buffer which you can then start typing text into.

Open Pops up a dialogue box allowing you to graphically select a file to edit.

<Alt-E> is the shortcut for this command.

Close Removes from memory the current file you are editing. If you have made changes to the file then you are prompted to save the changes first.

Insert file Pops up a dialogue allowing you to select a file to include in the current buffer.

Save Saves the current work. **<Alt-W>** is the shortcut for this command.

Save as Allows you to save the current buffer with a new filename, rather than use the one which you used when you selected the file for editing.

Save all Saves all modified buffers. It is quite convenient if you are making a lot of changes to a variety of buffers, and are delaying saving until all the files are synchronised.

Filter Region Pops up a dialogue box allowing you to type in the name of an external command. The current buffer is passed to the command and the result of executing that command is read back into the buffer.

File operations Submenu containing a command which allows you to reload the current buffer from disk. This allows you to discard any changes you have made (you are prompted before the file is reloaded).

Change directory Allows you to move around the filesystem so that it is easier to specify filename based on the current directory, rather than having to type long filenames. Not needed that often as you can achieve the same thing when you open a file for editing.

Print Used to select printing options before printing the current file.

Command Line Displays a command line prompt where you can enter commands not directly accessible from the keyboard or menus.

1, 2, 3, ... These menu items display the most recently edited files. These menu items provide a convenient and fast way to edit a file. You can customise how large this list is if you find the default of 6 entries too small for you.

Edit This menu contains functions to perform common editing tasks.

Undo

This command undoes the last action on the current buffer. There is an infinite undo buffer, so you can keep on undoing all the way back to the last file save - or beyond.

Undoing past the last file save has no effect on the file on disk, simply what CRiSP has in memory.

Redo

This command undoes the last undo command. You can access the same functionality by using this icon.

Delete

Use this command to delete the currently selected text. You can access this command with the <Del>key.

Insert Link

Used to insert a text-button in the current buffer which can be used to access reminders, macros or other documents. See the section on Links for full details on creating and using links

Insert Literal

Pops up a dialogue box allowing you to type in a character value to insert. This can be useful if you need to type in a character when you only know its numeric value but not how to type it on the keyboard, e.g. if its

a character from an international character set.

Toggle-Insert

This command toggles between insert and overtype mode. You can achieve the same affect by clicking on the INS or OVR panel in the status bar

Fill/Number

Used to access the Fill dialogue boxes which enable you to insert a column of numbers or fill a region with a selected pattern.

Indent & Unindent

These commands can be used shift right(indent) or shift left (unindent) the currently selected area. If no area is selected then the current line is affected only. You can normally achieve the same effect by using the <Tab> or <Shift-Tab> short cuts.

Transpose

Swaps the order of the two characters to the right of cursor and advances the cursor forward.

Text formatting

This accesses a sub menu containing various commands for formatting text, such as reformatting paragraphs or case conversion.

Columns…

Specify column width and perform various operations upon that column.

See also Sort > Column.

Macro start/stop

A keystroke macro is a recording of keystrokes, allowing you to repeat the keystrokes at a later time. This menu item is used to start a keystroke macro recording. If a keystroke macro is already being recorded, then this will terminate the definition.

Macro playback A keystroke macro is a recording of keystrokes. This menu command allows you to play back a previously recorded string of keystrokes.

View Use this menu to select actions which affect the way you see the current buffer, or to see extended dialogues showing you information about buffers, files keystroke macros, etc.

Find Various functions for finding strings in buffers and files.

Search for Displays a dialogue box allowing you to search for a string.

Search next Repeats the last search command. You can use <Keypad-5> or <Shift-F5> as a shortcut.

Replace Displays a dialogue box allowing you to perform an interface search and replace operation.

Search buffers

Lets you search all the currently laded buffers for every occurrence of a string.

Search files

Search files on disk for all occurrences of a string.

File replace Lets you perform a global search and replace operation on files.

File matching brace Move cursor to corresponding brace or bracket

Goto bookmark Allows you to jump to a previously saved bookmark.

Goto line Dialogue that lets you specify a line and/or column position to jump to.

Build tags Builds a new tag database.

Goto tag Jump to definition of a function.

Pop tag Return to previous position before the last jump to tag command.

Intelligent search Allows you to search for a string dynamically as you are typing it in.

Options Use this menu to access the CRiSP subsystem for changing personal preferences and settings, such as fonts, keyboard bindings, on-screen colours. There is a large collection of settings which can be changed to suit your taste. It is worth browsing through some of these options when you feel more comfortable with the product. Different people have different needs and tastes and it is here that you should be able to change how CRiSP works to suit you.

Autosave & backups
This allows you to configure preferences with regards to whether autosaving should be allowed, how frequently they are made; whether backup files are to be created and how many backup files to keep, and also other miscellaneous file oriented setup options.

Buffer Shows you the attributes defined for the current buffer.

Colour This option allows you to configure CRiSPs colour settings and allows you to configure colours for different parts of the CRiSP windows and also for how language specific keywords, etc., are displayed.

Cursor Allows you to configure the look and feel of the editing mode cursor. Select from an I-Beam, caret, block or underline cursor. You can configure different cursors for insert and overtype mode.

Font Allows you to configure the default font.

Key Bindings This allows you to re-map the keys on the keyboard to your own personal tastes.

Keyboard & Mouse

This presents a dialogue box with two tabbed sections allowing you to configure certain preferences about the keyboard and how the mouse functions. Keys allow you to redefine whether function keys pop up dialogue boxes or present a character-mode command line prompt; the Mouse option configures certain aspects of the mouse, e.g. whether to have Unix or Windows style behaviour; Options presents a list of various options, such as how line wrapping should work, virtual space settings, etc.

Language editing modes

This dialogue box allows you to define how CRiSP manipulates files with specific file extensions. CRiSP allows you to configure options on a per-file extension basis. You can configure such things as margins, Auto-wrap, whether language specific keywords are to be enabled, and so on.

Keyword Builder Dialogue box to add new languages to the Colouriser mechanism, or to add new keywords to existing language descriptions.

Memory configuration

This option allows you to tune how CRiSP uses memory when reading and editing files. Normally you will not need to use this option, but it can be useful under certain special circumstances, e.g. running CRiSP on small systems configurations when you want to edit huge files (files which are too large for RAM or virtual memory address space).

Also this option allows you to configure certain safety aspects of file editing - to avoid problems when two or more users edit the same file.

Miscellaneous This category is for options which do not neatly fit into any of the other categories. This includes defining a command for printing files, and setting up private start-up macro files.

Printer setup Allows you to configure how the printer is accessed, e.g. command for printing, 2-columns, PostScript support, page attributes.

Screen Settings This category is used for setting up various aspects of the screen. This includes enabling and disabling certain GUI parts of the screen (either scrollbars, menu bar, etc.), and what parts of the status line are to be displayed.

Screen Size A sub-menu of common and useful screen size settings.

Search & Replace Configure the default search options.

Source Code Control Lets you configure external commands for accessing the source code control system.

Start-up These options allow you to configure how CRiSP starts up, e.g. whether it restores files and window layouts, and also what visual interface components are to be visible.

Tag setup Configures the project and global tag files, and defines the parameters for how frequently dynamic tag updating of the class browser window is performed.

Toolbars Allows you to create and customise toolbars, and define which ones are visible on start-up.

Save setup Saves the current configuration to disk.

WWW Browser Configure the browser to be used for the WEB browser icon which loads the current file into the browser.

Tools Miscellaneous collection of tools, such as compiling programs or seeing programmer oriented information.

ASCII chart Displays a character set chart. This is most useful for programmers. Options are provided for viewing the character codes in decimal, octal and hexadecimal. You can invoke the same command by running the ascii macro at the Command: prompt (<F10>).

Change Log Utility macro to aid in creating edit histories for projects.

File diff/merge Dialogue box supporting commands to compare two versions of a file and to merge two files into one.

Mail GUI mail reading macro - only applicable to Unix machines.

Diff files Provides a way of comparing two files and marking the changes (differences) between them.

Operator hierarchy Dialogue showing operator precedence chart for C/C++ programmers.

Sort Dialogue box allowing you to sort parts of the buffer, including columns.

Check in,.. RCS History Commands for checking files into and out of a source code control database.

Compile Compile current file based on previously defined setup options.

Make Build project using the make facility.

Shell buffer Create a shell buffer, which can be used to capture output of a shell-process.

Spell check Perform a spell check on the current document, and/or set spell checker options.

Windows Used to manage CRiSP windows and dialogue boxes, such as selecting a hidden dialogue or iconising dialogue boxes. The most common function to be used here is the **New Window** option which creates a new editing area (also referred to as a *peel-off* window). This new window is controlled by the same copy of CRiSP and provides a more convenient way to edit multiple files in different windows. See online help for details.

Help Various entries into the on-line manuals.

Feedback Use this command if you need to report a bug or wish to send feedback on the product. Using this causes CRiSP to create a new buffer and includes various information about the version of CRiSP you are running which is useful to the support personnel.

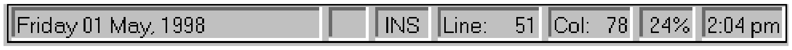
Key Bindings Provides a quick summary of the current function keys mapped.

Release notes Contains a summary of changes to CRiSP since the last few releases.

User Guide Contains detailed usage information on using CRiSP. As you become more accustomed to using CRiSP, it is worth browsing through this to find out about some of the more powerful features available.

The Status Bar

The status bar, located at the bottom of the CRiSP window is designed to provide summary information about the current file and editing environment. The status bar is divided up into a number of sections, some of which are clickable on to provide fast access to various facilities.



Friday 01 May, 1998 | | INS | Line: 51 | Col: 78 | 24% | 2:04 pm

The left most panel is reserved for status messages, e.g. menu help when moving the mouse through a menu.

The next panel shows the status of the macro recorder. If a keystroke macro is being recorded then the string **RE** will be displayed in this area. Otherwise it will be blank. You can record keystroke macros via the **<F7>** key and play them back via **<F8>**.

The next panel displays the current input mode, either **INS** if you are in insert mode, or **OVR** if you are in overtype mode. You can toggle between modes by clicking on this status panel area, or by using the keyboard shortcut **<Alt-I>**.

The next three panels show the current line number, column number and percentage of the way through the current buffer. Clicking on any of these will pop up the **Goto Line** dialogue box allowing you to move to a particular line and/or column number.

The last status panel shows the current time. If you click on this then the current time will be inserted into the current buffer.

Introduction to the Keyboard

Most of the time, you will interact with CRiSP using the keyboard. Occasionally you may use the mouse. The way you use CRiSP will depend very much on the type of work you are doing. If you are at the start of a brand new project, you will probably spend most of your time typing in and making simple corrections as you progress.

If you are fixing the odd bug in a program, or reviewing source code or data files, you will not be spending a lot of time typing in, but will probably spend more time browsing around - searching for things, and arranging the screen to give you a better view onto your data.

The keyboard is used for two types of activities: typing and editing, i.e. performing some editing command. Fortunately, most computer systems have now standardised on the IBM PC keyboard layout, making it easier to move from one system to another, even on non-Windows systems, such as UNIX.

The keyboard is divided up into a number of sections:

- the typeable keys, usually in a QWERTY layout (unless you are using a country specific keyboard),
- the numeric keypad, with associated function keys around the edge of the keypad,
- the cursor keys, including PgUp, PgDn, etc.
- Function keys along the top of the keyboard.

The typeable keys perform the obvious action of allowing you to type in plain text. Actions such as moving the cursor around can be done with the cursor keys. All the other keys on the keyboard are used to access various editing functions within CRiSP.

The keys on the keyboard that are labelled with actions, such as **Home,** and **End** perform the described action, and are summarised below.

In addition, each key can be pressed with a *modifier* key which provides access to far more keys than would be feasible on the keyboard alone. A normal PC keyboard has 102 keys (105 if you have one of the newer Windows/95 keyboards). Given that 50+ of these keys are typeable keys, this would only leave room for 40-50 editing functions. All keyboards come with a set of modifier keys, some of which are obvious, and others which may be new to you.

Shift Normally this key is pressed along with a letter or digit key to provide an upper case version of a letter, or to access some of the extended characters on the keyboard, e.g. exclamation mark (!), or ampersand (&). Whether the shift key works with the non-typeable keys is usually a function of the program you are using.

Ctrl Usually used in conjunction with the letter keys to generate control codes. Control codes have little meaning in today’s age of fast desktop computers and fast video screens. In older days, the **Ctrl** key was used to perform physical actions, e.g. on a line-printer or communications line. Most desktop applications use the **Ctrl** key to provide access to additional functions.

Alt The ALT key is a modifier key. On many systems, the ALT key is used in conjunction with the typeable letter keys to provide global functions. For example under Windows and Motif, **Alt-F4** is a standard key combination used to terminate the current application. The windowing system on your machine usually intercepts certain key combinations, but an application can make use of the remaining Alt-key combinations.

CapsLock This key allows you to lock the keyboard so that all typeable letters come out in upper case. Most software packages let you use this key without adding any extended meaning to the key.

All these keys are known as modifiers - they modify the meaning of the key you type. The **Shift** and **CapsLock** keys are hangovers from the old typewriter days, and when used in conjunction with the standard typeable keys, they provide access to the less frequently used punctuation symbols. CRiSP makes great use of the **Shift, Ctrl** and **Alt** keys when used in conjunction with the non-typeable keys. For example, **Ctrl-Left-Arrow** is used to modify the meaning of the **Left-Arrow** key. **Left-Arrow** on its own moves the cursor left one position. When combined with the **Ctrl** key, the cursor moves back to the start of the previous word.

As you can imagine, these key combinations give access to many more functions than would be achievable by just relying on the plain function keys.

Some key combinations you will use most frequently and you will learn and remember these quite easily. Others may be much rarer. The commands which are more obscure or rarer keystrokes are also assigned positions in the menu hierarchy or on the icon bar. There is usually enough ways to find a function, that you shouldn't feel too lost.

Other than the cursor keys, backspace and delete keys, the most frequently used function keys are the **Alt-X** keys, where **X** is a letter. These keys are designed to be easy to remember. For example, **Alt-E** is used to edit a file, **Alt-W** is used to write (save) the current file.

For commands which require further input, e.g. the name of a file to load, you will be prompted for the input. Depending on the command and the configuration settings, you will either be presented with a dialogue box, or be prompted on the command line. This caters for users who prefer dialogue boxes or the command line. In a dialogue box, you are presented with all the options and you can use the mouse to make selections. With the command line, you simply use the keyboard, thus avoiding having to move your hand from the keyboard to the mouse and back again.

Useful common keys

The following sections describe the most common keys and are organised by topic. This section provides a very brief summary of some useful keys which you may need to know to get started, or to exit from CRiSP.

<Alt-E>	Edit a file.
<Alt-X>	Exit from CRiSP. This prompts you if you have any modified but unsaved buffers loaded.
<Alt-W>	Save current file.
<F10>	Accesses a **Command:** prompt where you can type in commands which are not normally assigned to a function key, menu item, or icon.

Cursor Movement

<Left-arrow>	Move cursor left one character
<Right-arrow>	Move cursor right one character
<Up-arrow>	Move cursor up one line.

<Down-arrow>	Move cursor down one line.
<Ctrl-Left-arrow>	Move cursor left one word
<Ctrl-Right-arrow>	Move cursor right one word
<Ctrl-Up-arrow>	Scroll window up one line.
<Ctrl-Down-arrow>	Scroll window down one line.
<Home>	Move cursor to start of line. Second time you press this key, moves cursor to top of the window. Third press, moves cursor to start of file.
<End>	Move cursor to end of line. Second time moves cursor to bottom of window. Third time, moves cursor to end of file.
<Ctrl-Home>	Move cursor to top of current window.
<Ctrl-End>	Move cursor to end of current window.
<PgDn>	Scroll screen up down page.
<PgUp>	Scroll screen up one page.
<Shift-PgDn>	Move to next function.
<Shift-PgUp>	Move to previous function.

Typing and Deleting Text

<Backspace>	Delete character to the left of the cursor.
<Del>	Delete character to the right of the cursor.
<Ctrl-Backspace>	Delete word to the left.
<Alt-Backspace>	Delete word to the right.
<Alt-D>	Deletes the current line.
<Alt-K>	Delete text from the cursor to the end of line.
<Alt-I>	Toggles between insert mode and overstrike mode.
<Alt-U>	Undoes the previous edit operation
<Ctrl-U>	Redo - undo the last undo operation

NOTE

> On some keyboards, most notably UNIX machines on DEC equipment, the **Backspace** and **Del** keys will appear to be swapped the wrong way around. CRiSP provides a configuration option which allows you to swap these keys. See **Options->Keyboard & Mouse** if you need to change this option.

Typing in text is done simply by typing in the text. CRiSP supports an insert mode and an overtype mode. In insert mode (the default), text to the right of the cursor shifts up to make room for what you are typing. In overtype mode, you overwrite the text underneath the cursor. You can switch between these modes by using the **<Alt-I>** function key, or by clicking on the **INS** or **OVR** status bar panels

Selecting text using the keyboard

You can use the keyboard as well as the mouse to select a region. Sometimes, the keyboard may be easier than mouse, especially when selecting areas larger than the visible window.
CRiSP supports a number of styles of selection.
In all these cases, you press the key to enter the desired selection mode. You can then use the cursor keys to move around the buffer, and either hit one of the scrap keys (e.g. **<Keypad-Plus>**, or **<Keypad-Minus>**) or hit the selection key again to deselect the region. You can start a selection with the keyboard and continue using the mouse, or vice versa, depending on what is most comfortable for you.

Key	**Type**	**Meaning**
<Alt-A>	Non-inclusive	The non-inclusive marker is used to select text a character at a time. The character under (or to the right) of the cursor is not included in the highlighted selection.
<Alt-C>	Column	This key is used to select columns.
<Alt-L>	Line	Selects entire lines at a time.
<Alt-M>	Normal	This is similar to using the mouse, and also similar to <**Alt-A**>. The character under the cursor is included in the region.

Using the Mouse

CRiSP supports 2 and 3 button mice. The normal user interface assumes that you have a 3 button mouse under UNIX versions of CRiSP and a 2 button mouse under DOS-based PCs (e.g. Windows/3 and Windows/NT). On these systems CRiSP treats two button mice as if they were three button mice but with the middle button missing.

The mouse is used in a number of ways. You can click on different user-interface objects to cause something to happen, e.g. the status bar or icon bar, or you can select a region of text before applying a command to it, i.e. dragging over the text buffer.

Clicking Clicking involves pressing one of the mouse buttons and releasing the button. The mouse pointer should not move during this press-release action (otherwise you will be performing a drag action). Different mouse buttons may produce different effects depending what you click on.

Types of Clicking You can normally perform single, double or triple clicking. For example, pressing the left mouse button, releasing, pressing again, and releasing again is a double click. The difference in behaviour for single, double or triple clicking will depend on what you click on. Normally, single clicking is used, for example, in a list selection box to highlight the entry you are clicking on; double clicking is a GO action telling CRiSP you are happy with the item selected and to perform the appropriate action.

Selecting text Selecting text involves pressing the left mouse button down, and whilst the button is held down, moving the mouse cursor. Dragging is used mainly in the editing area of the CRiSP window to highlight pieces of text.

Dragging Dragging is the ability to select a region of text and then drag it to somewhere else where it is moved to. To do this, you select the region, and then click inside the selected area and drag the mouse cursor to the position where you want the text to be moved to.

The actions you can do will depend on the type of object you are clicking on. Many aspects of the CRiSP user interface will be familiar to users of graphical windowing systems.

Mouse Action in the main CRiSP window

The least obvious area of the CRiSP user-interface is the main editing window, where clicking on different parts of the window, with different mouse buttons, gives you various functions. These functions are:-

Left Single click This will **move the cursor** to the point clicked on for example to quickly switch between buffers being viewed in multiple windows.

Left Double click On a word will **highlight the word**.

Left Triple-click On a line will **highlight the current line**. (i.e. the same as using <Alt-L> on the keyboard)

Dragging Dragging the mouse after a single, double or triple click will **highlight a region**. When dragging after performing a double or triple click, CRiSP will highlight only full words or complete lines, respectively. Thus a triple click is a convenient way to highlight interactively a group of lines, in a similar manner to using the **<Alt-L>** key from the keyboard.

You can force the window to scroll so that you can highlight more of the editing window than is visible simply by moving the mouse pointer outside of the CRiSP editing window and continuing to move the mouse.

Drag and drop text moving This facility allows you to select a region of text and then *drag* or move it to another place in the buffer or in another edit window. To do this, select the region, and then click and hold the left mouse button on the selected area and move the mouse to the destination. Release the mouse to move the text.

If you want to abort a drag operation, release the mouse whilst it is inside the selected area.

Control If the left mouse button is pressed with the **Control** key held down you can **highlight a columnar region** (i.e. the same as using the <Alt-C> key from the keyboard).

Shift Using the shift key after highlighting a selection and moving to a new point will **extend the highlighted region** to the new position.

Middle-button Using the middle mouse button after selecting a region will **paste the selected region** at the insertion point and delete the original region. So the first press of the middle mouse button will appear to have no visible effect. Pressing the middle mouse button again will paste the same selected region at the insertion point. You can continue to paste the text as many times as required by pressing the middle mouse button. (*Note - it is possible to amend the way the middle mouse button works by using the Setup option “Auto-region delete)*

Right Mouse button

This pops up a menu from which the user can perform **common editing tasks.** These tasks are File, View, Edit, Find and Extras. This popup menu stays on the screen only as long as the right mouse button is pressed down - you can select any of these options or you can abort the menu by releasing the right mouse button outside of the area of the popped up window.

Mouse Action on a Window Border

Single click left button

A single click with the left mouse button on a window border will *split* the window, making it into two. If the mouse is too near an extremity of the window such that splitting it at that point would create a window too small to be visible, then nothing happens. The best place to split a window is normally in the centre (but not on the title of the window - see below). Attempting to split a window where a split already occurs allows you to move the window boundary.

Dragging

Pressing the left mouse button and keeping it down whilst dragging will allow you to split a window and more accurately position the window split. Releasing the left mouse button will confirm the position of the split. Note, that window splitting can appear a little *quirky* in certain situations owing to ambiguities which may not be apparent at first. E.g. attempting to split a window where a split is already there may have no effect. It is best to experiment with this facility. [This functionality is implemented using the macro language and may be extended fairly easily]

Clicking on the Title (buffer name) of a window

Clicking on window title

May be used to select the *next* buffer, as if the **<Alt N>** key had been pressed. (Note that clicking the right mouse button does **not** select the previous window, since it is reserved for bringing up the popup window described above).

Clicking on the Time indicator on the status line

Clicking on status line	This will insert the current date into the buffer. Clicking with the middle mouse button will insert the current time.

Copy, Cut, Paste

CRiSP provides a cut and paste facility. The cut and paste facility allows you to mark a region of text and copy it to another part of the same buffer, or to another buffer. The process of marking this text is referred to as *selecting*. A selected part of the buffer is shown as highlighted text on screen.

Selecting text and pasting it somewhere else is such a common editing action, that CRiSP provides numerous ways of achieving the same thing, e.g. you can use the keyboard, mouse, menu entries or icons to achieve the same result.

When text is to be copied to some other buffer or place, it is first copied to a temporary work area, called the **scrap**. The scrap is a special buffer which holds the text to be copied until it is needed. This allows you to mark text ready for pasting somewhere else, but allows you to delay the act of pasting until you are ready for it.

How to invoke cut and paste operations

Keyboard - selecting region

Key	**Type**	**Meaning**
<Alt-A>	Non-inclusive	The non-inclusive marker is used to select text a character at a time. The character under (or to the right) of the cursor is not included in the highlighted selection.
<Alt-C>	Column	This key is used to select columns.
<Alt-L>	Line	Selects entire lines at a time.
<Alt-M>	Normal	This is similar to using the mouse, and also similar to **<Alt-A>**. The character under the cursor is included in the region.

Use the cursor keys to extend the region.

Keyboard functions

<Keypad-Minus>	Cut selected area to the scrap

<Keypad-Plus>	Copy selected area to the scrap
<Del>	Delete selected area.

Selecting using the mouse

Selecting using the mouse allows you to define the region of text that is to be affected in a copy or cut operation. You can use the keyboard to achieve the same effect. Sometimes it is more convenient to use the mouse, and at others to use the keyboard.

The following screen image shows what a selected, or highlighted region looks like.

You select an area using the mouse, by a process called *dragging*. Dragging involves holding the left mouse button down, whilst moving the mouse to sweep over the text to be selected.

CRiSP allows you to select text in character mode, word mode, line mode or the entire buffer, by clicking on the mouse first.

Click then drag	Selection is highlighted in character at a time mode.
Double click then drag	Selection highlighted a whole word at a time.
Triple click then drag	Selection is highlighted a whole line at a time.
Quadruple click	Entire buffer is selected.

Extending the selection

Rather than dragging with the mouse, you can use an alternate method. If you highlight a small area with the mouse, you can then extend the selected area by moving the mouse to the end of the desired area, and pressing the **<Shift>** key and the left mouse button. The existing selected area is then extended to the pointer where you pressed shift-left-mouse button.

Column selection

You can select a columnar region of text by holding down the **<Ctrl>** key whilst dragging. (Do not use double, triple or quadruple whilst selecting a column).

Manipulating the Scrap

Copying an area

Click on the Copy icon (or select Copy from the Edit menu). The selected area will be stored in a buffer until you are ready to paste it to a new point in the file (or to another file). The original area you copied will remain unaffected by this process.

Cutting an area

Click on the Cut icon (or select Cut from the Edit menu). The selected area will be deleted but will be stored in a buffer until you select another area.

Pasting an area

After selecting an area and using either Cut or Copy you can paste the selected area by moving the cursor to a new position and using the Paste icon (or selecting Paste from the edit menu).

Cutting or Copying to a named scrap

When you cut or copy an area, CRiSP saves it to an anonymous scrap buffer. In some situations e.g. editing a complex file, you may wish to save it to a named scrap file. This will enable you to retain multiple scrap files and to use them repeatedly during the current editing session and in later editing sessions. In this way you can build a personal glossary of useful scrap files to use when you are editing.

To use a named scrap file

Copy to named scrap **<Ctrl-Keypad-Plus>** and you will be prompted for the scrap file name

Cut to a named scrap **<Ctrl-Keypad-Minus>** and you will be prompted for the scrap file name

Insert a named scrap **<Ctrl-Keypad-Ins>** and you will be prompted for the name of the scrap file you wish to insert.

Viewing a list of existing scrap files

Press **<Tab>** after using any of the above options and you can view a list of your current named scrap files.

Appending multiple selections to one scrap file

For complex editing sessions you may wish to select several different areas, copy or cut them into a single scrap file so that they can be inserted as one complete section elsewhere. You can do this across multiple buffers and you can save the results in a named scrap file if required.

To copy and append to a scrap file

Select an area then press **<Shift-Keypad-Plus>.** Now select the next area you wish to add to the same scrap file and press **<Shift-Keypad-Plus>** again. Repeat this until you have selected all the areas you wish to include. The areas need not be in the same file but you should select them in the order in which you wish them to appear when they are eventually inserted.

To cut and append to a scrap file

Select an area then press **<Shift-Keypad-Minus>.** Now select the next area you wish to add to the same scrap file and press **<Shift-Keypad-Minus>** again. Repeat this until you have selected all the areas you wish to include. The areas need not be in the same file but you should select them in the order in which you wish them to appear when they are eventually inserted.

To paste the results of your appended scrap file

Position the cursor where you want to insert the text and press the paste icon (or use **<Ctrl-Keypad-Insert>**).

Editing Files

CRiSP is a plain text file editor. This is useful in those situations where a word-processor may not be appropriate, e.g. for creating plain text files, program source code, data files, etc.

CRiSP lets you edit multiple files simultaneously - you can load up and view or edit as many files as you like. Each file is stored in temporary memory, known as a *buffer*. These buffers are where your changes are made. When you have finished editing a file, you can save the changes back to disk. Loading multiple files and editing them at the same time means you do not have to finish with one file before moving onto another file to edit.

There are a number of ways to load a file into memory for editing. You can specify filenames on the command line, or use any of the following:

<Alt-E> keystroke
Click on the **File->Open** menu option.
Click on the open file icon. 📂

By default, CRiSP pops up an open file dialogue box. The actual appearance is slightly different depending on the operating system you are using. Under Windows/95, Windows NT/4.0 or above, you will get a familiar Explorer style dialogue box. Under UNIX, you get a dialogue box similar to the Explorer.

In all cases you can navigate around the file system hierarchy and select files. When you select the **OK** option, the file or files you have selected will be loaded in CRiSP. The dialogue box contains various options that let you select to edit a file in read-only mode (in case you do not want to make any accidental changes), and to select the file type, e.g. edit a UNIX file when on a DOS system, or a DOS file on a UNIX system.

You can select multiple files to load by using the **Shift** key in conjunction with the left mouse button when making selections.

File Manipulation

<Alt-B>	Pops up a dialogue box showing you all the files loaded into CRiSP and making it easy to select amongst them.
<Alt-E>	Edit a file
<Alt-R>	(Read) Insert the contents of another file into the current buffer at the current cursor position.
<Alt-W>	Write the current file (buffer) to disk.
<Ctrl-Minus>	Deletes the current buffer from memory. Prompts you if you have not saved your modifications.

Buffers and Files Overview

CRiSP is used to edit files stored on disk. Whilst you are editing a file, it is stored in an area called a buffer. Each file you edit is stored in its own private buffer. The buffer provides a mechanism where a user can make changes without affecting the copy of the file stored on disk. Changes to a buffer can always be "undone" using CRiSP’s *undo* feature. During normal editing, there are 3 or more *versions* of a file in existence. These multiple versions allow the user to edit a file at speed and be safe knowing that the original file on disk will not be overwritten until committed explicitly by the user. These multiple versions are explained below.

Buffers and Windows

CRiSP allows you to view buffers in a window. There are two types of windows - internal character mode windows, and top-level edit windows. Top level edit windows are characterised by having its own toolbar, menu bar, etc. You can create new top level windows via the **Window➜New Window**. Character mode windows can be created using the **<F3>** function key or by using the mouse on window borders. It is important to note that CRiSP allows you to see the same buffer in different windows, and that any changes made in one window may affect what you see in the other window(s). If you want to edit a file in two windows with each window having its own independent copy of the file, you will need to run up another invocation of CRiSP.

CRiSP allows multiple buffers to be loaded during a work session. This is useful because a user can wander into a large directory and just use the wildcards to specify collections of files on the command line or at the Edit File prompt.

You operate on one buffer at a time, and can switch buffers at any time, e.g. by clicking in the window of interest or using the window manipulation function keys. Different windows can display different buffers or even the parts of the same buffer allowing you to see two or more parts of a file. (Switching buffers is normally accomplished with the **<Alt-N>**, **<Alt-P>**, or **<Alt-B>** keys. The first two switch to the next and previous buffers, respectively. The latter displays a list of buffers currently loaded).

There is no limit to the number of files which can be loaded into CRiSP at any time, except those imposed by the system you are running on, e.g. the amount of memory or temporary disk space consumed.

CRiSP is very careful never to accidentally lose the users work, whether due to a user error or a hardware problem (such as losing power to the machine). This is achieved in a number of ways, e.g. autosaving and creation of backups. These mechanisms are described below.

You can make changes to the copy of a file loaded into the buffer. This copy is only temporary and all the information is lost if a power-fail occurs. When you try to exit from CRiSP, CRiSP reminds you if any modified buffers exist and gives you a chance to save the changes or cancel the exit. When a modified buffer is saved away to disk, a backup is made of the original file, so you can always revert back to the state of the file before the CRiSP editing session started.

Autosaving Files

CRiSP provides an autosave mechanism which involves saving any changed buffers back to disk (without overwriting the original file), either after so many keystrokes or after a default period of inactivity. (The default is 1 minute). This feature is very valuable. Frequently you might be in the middle of editing a file only to be called away (for that urgent cup of coffee, perhaps, or to answer the phone). You may even go out to lunch or have the CRiSP editing session obscured (e.g. iconised in a GUI environment) and totally forget about saving the file. The autosave facility makes a copy of the modified files, as if you had saved the file, but without overwriting the file.

CRiSP normally creates autosave files by prefixing and suffixing an ‘at’ sign(**@**) to the original file name; these autosave files are stored in your backup directory, or the current directory, if none is defined.

CRiSP automatically deletes these autosave files when CRiSP is terminated, or when you close the buffer. If your system crashes, you can look in your backup directory for copies of these files.

Making backups of files

When you save files, CRiSP creates a backup of the original file. By default this is stored in the same directory as the original file, but with a **.bak** file extension. You can configure a default backup directory so that all backup files are stored in one place, which avoids littering your source directories with backup files, and makes it more convenient to delete these files in one go.
Use the **Options -> Autosave & Backups** menu to access this option.
If you enable this option, then you can also tell CRiSP to create multiple backup files, i.e. you are not restricted to just the current file and a single **.bak** file. When you configure multiple backup files, CRiSP creates sub-directories in your backup directory, called **0/**, **1/**, **2/**, and so on. When CRiSP goes to create a new backup file, it moves the backups into the 0/ directory. What was in the 0/ directory gets moved to the 1/ directory and so on. The most recent backup file is always in the backup directory, and the next most recent file is in the 0/ directory.

Searching and Replacing Text

CRiSP provides numerous functions for searching for strings in buffers and files. CRiSP lets you do the following:

Search for a string in the current buffer
Show all functions, sections or routines in a file~~.~~
Show all lines that contain a particular pattern in the current buffer or all buffers
Show all lines that contain a pattern in files on disk
Search for and replace a string in the current buffer
Search for and replace all strings in all buffers or files on disk

Searching is performed by specifying a string to match. You can specify an exact string to look for or for a pattern (known as a *regular expression).* A regular expression is a way of describing inexact string matches. For example, say you wanted to search for all words which started with an uppercase letter. You couldn’t list every conceivable word, so you would use a pattern which allows you to match all words which have a similar look and feel.

When replacing a matched string, you can use special characters to perform such actions as uppercasing a matched word.

Introduction to Regular Expressions

A regular expression is a technical term used to describe a string of characters which can be used to match arbitrary patterns. Searching is such a common action that CRiSP provides numerous ways to achieve the same or related functions. The regular expression is what gives the search facility a great deal of power.

There are various aspects to a regular expression, but the thing that is common is that special characters are used to represent different types of patterns to match.

A simple example of why you would need to use a regular expression is because you might want to find all occurrences of the word **Politician** but only if it is followed by the word **Government** somewhere later in the line. A simple regular expression to do this would be:

```
        Politician.*Government
```

The interesting part of this line is the magic characters dot and asterisk which together mean to match any sequence of characters.

For historical reasons, CRiSP supports two sets of regular expressions, known as CRiSP syntax and UNIX syntax. The CRiSP syntax is aimed at compatibility for user's of the MS-DOS based editor, *BRIEF*. UNIX syntax is the common form of patterns used by various UNIX utilities. You are advised to stick to UNIX regular expressions whenever possible. Both UNIX and CRiSP syntaxes are equally powerful, it is just that they use differing magic characters to denote the patterns to match.

Regular expressions start off with some very simple ideas and concepts. The power of regular expressions is that by combining the various operators (special characters) available, you can end up with a very powerful string that can perform context sensitive matching. Unfortunately, the more complex a regular expression, the more difficult it is to decipher on a first reading, and long regular expressions are not that different looking to garbage.

The trick with using and writing long regular expressions is to build them up a piece at a time. Regular expressions can be used for such operations as swapping two columns of floating point numbers. The expression to do this is not necessarily obvious, but is actually quite easy to do a piece at a time.

Regular expressions are available not only as a one-time function whilst manipulating a file, but also from the macro language.

The thing to remember when trying to use regular expressions is to not be too over ambitious. Build up a regular expression a piece at a time, checking to see what it matches, and if the string matches things you do not want it to, then to add more sophistication to the regular expression until it does exactly what you want.

^ and **$** The ^ character in an expression is used to match the beginning of a line. The $ character is used to match the end of a line.

. (Full-stop or dot) is used to match any single character.

+ and ***** + after an expression, e.g. ".+" means to match **one or more** of the specified expression. * is used to match **zero or more** of a pattern. For example, to match one or more digits, you might use a pattern such as [0-9]+

? Search for **zero or one** occurrences of the preceding expression.

\< and \> The \< sequence is used to match the beginning of a word. It is a very useful expression, and avoids trying to write an equivalent regular expression using the other facilities available. Without this, matching a word is difficult, because you would need to take into account words starting at the beginning of a line (i.e. no characters preceding it) and would need to avoid matching inside the middle of a word.
\> matches the end of a word.
Thus, **\<fox\>** matches the word **fox** but not the word **fox** appearing in **foxtrot.**

[...] A sequence of characters inside square brackets matches any of the characters within the brackets. For example, [A-Z] matches any upper case letter. You can list all the characters to match or use a dash to separate a range of characters. E.g. to search for all even digits, you could use [02468]. If the first character after the open square bracket is a ^, then this means to match any characters not listed in the brackets. If you want to include a dash or close square bracket character in the list, then place them immediately after the open bracket.

\{n\}, \{n,m\} Use these constructs to search for a sequence of matching expressions. The first example searches for exactly **n**

matches of the preceding expression. The latter example searches for any of **n, n+1, n+2, ..., m** of the preceding expressions. For example

[0-9]\{6\}

would search for exactly 6 digits.

\(...\) Use this to group a sequence of regular expression patterns. This is useful when combined with one of the repetition operators (e.g. *, +, ? \{..\}) or when performing a search and replace operation, in which case it allows a matched string to be inserted as is.

\c This isn't really a regular expression matching operator. You use this inside a search string, to say that when the search completes, to leave the cursor positioned at this point. Normally the cursor is left at the start of the matching expression (which is normally highlighted).

For example, suppose you had a database, consisting of lines containing a number, followed by a colon, followed by some text:

```
0000:Record one
0001:Record two
...
```

You might want to find a particular record, but leave the cursor position just after the colon. You can do this with the following regular expression:

```
^[0-9]+:\c
```

\m This is a special character used to find a **marked** line. Marked lines are created when you use the file difference facility. By looking for this expression you can quickly find the lines that are different compared to the other file.

\n Use this to match a line boundary. The **\n** sequence allows you to match strings which start on one line, and extend into a succeeding line. The **$** regular expression characters matches the end of a line, but can only be used at the end of a regular expression. In fact, the **$** character doesn't include the newline at the end of each line in the match, so you would need to use **\n** to include the newline character.

\t This is an alternative notation for a TAB character. It is most convenient when writing macros, as it is easier to then distinguish a tab from a space.

\xNN	The sequence **\x** followed by two hexadecimal characters allows you to search for any arbitrary character, whose value is specified by the hex characters. This is useful when looking at binary data, or when you need to type characters with the top bit set, and you do not have a key on the keyboard which can type the key.

How to invoke Search and Replace commands

Keyboard

<Alt-S>, <F5>	Search forwards for a string.
<Alt-Y>, <Alt-F5>	Search backwards for a string.
<Shift-F5>, <Keypad-5>	Repeat last search.
<Alt-T>, <F6>	Search and replace
<Ctrl-6>	Find matching brace
<Ctrl-F5>	Toggles case sensitivity mode.
<Ctrl-F6>	Toggles regular expression mode. When this is turned off, the normal regular expression characters do not apply, and can be typed in without having to use a backslash character.

Command prompt macros

grep	Search files on disk for lines matching a pattern.
search_all	Search all buffers for lines matching a string.
search_one	Search current buffer for all lines matching a string

Icons

Search	Repeat search	Search & Replace	Search buffers.

Windows and viewing files

Although CRiSP is an editor for various types of files, you could also consider CRiSP as a file *viewer*. You will probably find that you spend at least half your time simply viewing files, trying to absorb their informational content.

As described elsewhere, CRiSP provides numerous facilities for editing and modifying files; but one of the more valuable uses for CRiSP is in correlating the data contained within files. Within this scope, you are provided with facilities for creating multiple views (windows) onto files, and for getting summary information on the files.

Windows for editing and viewing buffers

CRiSP is a multi-windowed editor. This means that the user can have different views of a buffer on screen at the same time There are a number of different ways to achieve this:

> character mode (or internal) windows with a single top level application window
>
> multiple top level windows
>
> combined character mode and top level windows

You can create and change the layout of windows as and when necessary, to give you either an expanded view of a file or to see multiple parts of the same or different files at the same time. One window is always referred to as the *current* window. This is the window where the flashing cursor appears and where your editing changes are made. There are no limits on the number of windows open at any one time.

It is important to realise that CRiSP uses buffers as a temporary storage area to allow you to edit files. A buffer is viewed through one or more windows. This mechanism allows you to see different parts of the same buffer at the same time.

Character mode windows

When CRiSP initially starts up, you will see a single character mode window occupying the entire editing area. At the top of the window is the window title, which usually corresponds to the filename of the buffer you are editing.

Character mode windows are convenient for editing and most of your time you can do all your editing within the confines of a single top level window. CRiSP provides numerous functions for moving, splitting and creating character mode windows, using the mouse or the keyboard.

The primary function keys for manipulating character mode windows are:

Key	**Mouse equivalent**	**Function**
<F1>	Click inside window	Select a new current window.
<F2>	Double click and drag on window edge.	Move boundary between two windows.
<F3>	Double click on window edge	Split window horizontally or vertically.
<F4>	Double click on window edge with middle mouse button.	Merge two windows together.
<Ctrl-Z>	None	Zooms or unzooms the current window. If you have multiple character mode windows on display, then the zoom feature hides all the other windows, and makes the current window full size. Pressing **<Ctrl-Z>** again unzooms the window restoring the hidden windows. This is a very quick way to get a larger view on to the current file.

The **<F1>, <F2>, <F3>** and **<F4>** keys require you to specify a border between two windows. For example, when splitting a window, using **<F3>** you need to indicate how the window should be split - vertically or horizontally, and whether the new window is above or below the current window. Use the arrow keys with these commands, or press **<Esc>** to abort the command.

Top level Windows, or Peel-offs

Top-level windows give you an alternative way of manipulating files. Using the top-level windows tends to require more mouse activity since it may not be possible to switch from one top-level window to another using the keyboard. Under Microsoft Windows, you can use **<Alt-TAB>** to cycle through top level windows, but if you have many applications, then this can be tiresome.

The advantage of top level windows is that to some extent, you can manipulate the layout of the internal character mode windows in one top-level window, without affecting any of the other top level windows on display. For example, you might spend most of your time using a single top level window, defining layouts for files and buffers, and then you get interrupted and need to look at some other, possibly unrelated files. You could initiate a new CRiSP session and load up the files, or you could just create a new peel off window, without disturbing your current layout, closing it down when you have finished.

It is important to realise that when you have two or more peel off windows, that these windows and the loaded files, are all under the control of a single copy of CRiSP. In fact it can be difficult to distinguish one copy of CRiSP controlling multiple peel off windows from multiple copies of CRiSP controlling a single top-level window each.

The normal way to exit CRiSP is via the **<Alt-X>** key or from the **File ➜ Exit** menu option. When you exit the last peel off window, CRiSP will prompt you to save any modified files which have not been saved to disk. When terminating a peel-off window, no such prompt is displayed. This can be a little disconcerting at first, but is entirely natural. CRiSP will only prompt you at the point where you might lose some valuable work. Creating and closing peel off windows is designed to be easy and troublesome without getting in the way.

You can even edit the same buffers in different peel off windows. You will see that both windows update at the same time, e.g. as you insert or delete text.

Top level windows are created with the **Window ➜ New window** menu item.

Saving the screen layout

CRiSP normally starts off with the editing window occupying the whole screen (on character based terminals) or the size of the GUI window (for the GUI versions of CRiSP). In the GUI versions of CRiSP, you can use the mouse to resize the window to claim more screen real-estate for the editing session, or the various window function keys can be used for splitting the current window.

CRiSP remembers the size, position and layout of your windows when you exit, so that they can be restored when you restart CRiSP. This feature is configurable (see **Options ➜ Start-up** for details on changing these defaults).

CRiSP also provides the projects facility for saving layouts of windows. This is useful if you continuously change the set of windows and buffers you are working on, e.g. if you work on multiple development projects and need to quickly switch from one set of files to another.

The Windows Menu

The windows menu, available at the top of the screen, contains a variety of functions, some of which have already been discussed.

The windows menu provides various facilities to help in managing your desktop. CRiSP provides numerous popup dialogue boxes, as well as the ability to edit in multiple top-level windows. CRiSP tries to be *modeless* meaning that you do not necessarily have to finish using one dialogue box before you can launch another. This means that after a while you can end of with a very untidy screen which makes it difficult to find one dialogue box or another.

Normally, when you invoke a facility, if that dialogue box is already present on screen (but possibly buried under the plethora of windows), then CRiSP will raise that dialogue box, so that you can see it (and also give it the input focus).

This selection of menu commands provides a control centre for manipulating windows.

Some of the entries are not actually specific to window management and these are described elsewhere:

Projects This provides a means for creating collections of windows and files to be edited together.

Scroll lock This provides a facility to have two character-mode windows scroll together at the same time. Useful when performing file comparisons (See also the file difference.)

Toggle borders Allows you to enable or disable the character mode window borders.

Zoom/unzoom Allows you to have just one character mode window in the current main window.

Iconize all

This entry is used to iconize all windows other than this one. This only applies to the dialogue boxes and edit windows belonging to this CRiSP session. Other applications or invocations of CRiSP will not be affected.

This is a very convenient command when you have lots of dialogue boxes or edit windows visible, or multiple copies of CRiSP running.

New window

This command creates a new editing window, complete with its own menus, toolbars, etc. This window is still controlled by the same copy of CRiSP used to launch it. Having multiple top-level edit windows can be convenient as it allows you to control window layouts, positions and sizes without having to spend a lot of time manipulation and re-sizing windows.

Note that this is the same copy of CRiSP controlling multiple windows. Any changes you make in one edit window will affect any other edit windows with the same file being displayed.

If you want to edit the same file independently, then you can launch multiple copies of CRiSP (although then you may be plagued with one copy of CRiSP telling you that someone else has changed the files you are currently editing).

More Windows

The menu entries labelled 1, 2, 3, ...6 provide fast access to the most recently created windows. You can un-iconize and/or bring to the top of the screen, the specified window. If there are too many windows to list in the menu, then a dialogue box can be used to make window selections, via the **More windows** entry.

You can either double click on a window name to select it, or use the **Select** button.

The **Iconize** button can to iconize a selected window.

Projects Facility

The projects facility is a way of associating groups of files for an edit session. This means you can load up a group of files in one go, and restore the screen layout, without having to reload each file individually.

This is similar and based on the way CRiSP stores the files you are editing when you exit, so that when you re-invoke CRiSP, it restores all the files and windows you were working on.

The projects command pops up a window showing you the currently defined projects, together with some status information about each project. Initially you will be presented with a blank set of projects.

You can give each project an arbitrary name which can be used as a convenient reminder of what files are in that group.

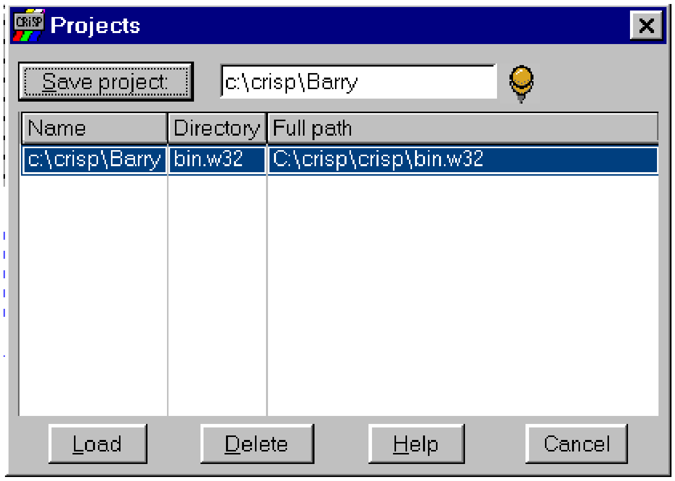
CRiSP associates the following information with each project:

Name	Each project has a name so you can easily remember which is which.
Working directory	Which directory you are in.
Files	The collection of files and buffers loaded into CRiSP.

Window positions CRiSP remembers the character mode window positions.

The top of the projects dialogue box allows you to type in a project name. If you are saving the current collection of files, then click on the **Save project** button.

If you have already loaded an existing project, then the current state of that project will automatically be saved when you load another project.



c:\crisp\Barry

Name	Directory	Full path
c:\crisp\Barry	bin.w32	C:\crisp\crisp\bin.w32

Next to the project name is a push-pin toggle button. By default the pin is in. If the pin is out then as soon you select a new project to switch to, the dialogue box will be dismissed.

You can double click on an existing project name to reload that project, or you can use the **Load** button.

The **Delete** button can be used to delete unwanted projects.

Restrictions

CRiSP will not restore a project if multiple top-level editing windows are currently being used. You will need to close each top-level window manually.

Scroll locking Buffers

Scroll-locking is a facility which you can use in one of two ways:

> You can use it to anchor the cursor so that it stays in the same position on screen, forcing the screen to scroll as you move the cursor, rather than just moving the cursor, or

You can use it to lock two windows together, e.g. when you want to scroll corresponding parts of two buffers, because for example, you are comparing two files.

Scroll-Lock on a single window

If scroll-locking is turned on, but only one character mode window is being displayed, then single-lock mode is enabled. In this mode, as you move the cursor, the window contents scroll around attempting to maintain the position of the cursor on screen.

This facility can be used when searching in a buffer, as it enables the line with the cursor on it to stay in the same relative position on screen, making it easier to spot where matching has occurred.

Scroll-Lock two windows

If two or more character mode windows are on display in the current main edit area, then activating scroll-lock will cause a prompt on the status line, asking you to select the window to be locked. (If only two windows are visible then CRiSP will use the *other* window).

With scroll-locking enabled, as you move up or down one window, the other window will move with it.

In this mode, the original window is referred to as the master window, and the *other* window is the slave window. As you move around the master window, the slave window will move as well. However, the reverse is not true.

Printing

CRiSP supports printing of files, optionally adding line numbers and titles to each page. The key useful feature of printing from within CRiSP is that the on-screen colorization can be transferred to the printed page, by suitable use of shading and coloring.

Printing works slightly differently under Microsoft Windows, compared to UNIX. UNIX does not contain a standard printing mechanism and so each application supports its own mechanisms to enable printing. CRiSP supports printing using the native printing subsystem under Windows; under UNIX a different mechanism is used.

If you are using UNIX and want to print to the standard print spooler, you can use the vertical pipe symbol to prefix the name of the command to send the printed output to. See below for examples of the "**Print to file:"** command.

Printing under Microsoft Windows

When you print a file under Microsoft Windows, the normal mechanism would be to click on the printer icon or use the **File ➜ Print** menu option. This invokes the standard dialogue boxes which let you configure the various printer options.

Under most circumstances, this will be all you need to do. Some user's may need to tweak CRiSPs printing options in order to create better printouts. For example, you may want to change the default font size for printing, change the number of lines per page or columns of printed text, or access the color PostScript options.

If you are not happy with the standard printing options, then you will need to change some of the options in the **Options ➜ Printer** dialogue box. See below for these options.

Printing under UNIX

UNIX does not contain the mechanisms for high quality printing. UNIX does provide a print spooling mechanism which is relatively straightforward to use, but an application cannot inquire about the type of printer attached, and so it is necessary to configure CRiSPs setup options in order to get the desired results.

CRiSP supports printing in a plain text format, suitable for all printer types, but lacks the facility to print in colour and bold, which is needed if you want to see colourised text buffers on paper as you see them on screen.

Invoking the Printer

There are a number of ways to invoke the printer:

1. Use the **File ➜ Print** option to print the current buffer.
   Under UNIX, you will be shown a dialogue box similar to the one shown below, allowing you to change options temporarily for this file to be printed. Press **<Enter>** to accept the default options and have the current buffer printed.
   Under Microsoft Windows, you will be presented with a dialogue box suitable for the default printer you have chosen.
2. Use the printer icon 🖨.
   Under UNIX, the current buffer will be printed using the options you have previously defined, without any dialogue box popping up. This is a quick way to print.
   Under Microsoft Windows, this option is the same as selecting the **File ➜ Print** option, and a dialogue box will be popped up.

3. Press the **<PrtSc>** function key. This key should not be used under Microsoft Windows, as it has a meaning outside of the scope of CRiSP (it usually means to create a screen dump).

   Under UNIX, this key is the same as using the print icon.

   If this key has no effect under X Windows, then you may need to configure the keyboard properly so that CRiSP knows that the key has been pressed. On many UNIX systems, this key is not defined properly, so you may see nothing happen when you attempt to print. If this happens to you, use the print icon or print menu item.

Columnar Text Manipulation

CRiSP provides a number of functions for performing columnar manipulations of data. This can be useful in all sorts of cases.

In order to make use of column operations, you first need to select a column from the current file. Selecting a column can be done using the mouse, or the keyboard. You can only select one column at a time (any attempt to select a second column will automatically unselect the first one).

A column can be as wide as necessary and as long as necessary. (A column does not necessarily extend from the first line in the file to the last line).

Column operations are useful to both programmers and non-programmers alike. Programmers can find it useful when dealing with repeated patterns of text (such as in #define statements). Non-programmers can find it useful when dealing with data files, e.g. generated files containing columns of numbers. These data files may need to be preformatted to fit some conventions so they can be used as input to some secondary program.

The following sections describe various column features:

Selecting	How to select a column of text
Cut and Paste	How to manipulate a selected column.
Column operations	Dialogue box for manipulating columns, e.g. to reformat numbers, or perform simple maths.
Column width	Dialogue box for moving text around inside a column.

One point to bear in mind when using the column operations: CRiSP is essentially a plain text file manipulation tool. It provides functions for manipulating columns, but it most certainly is not a spread-sheet. What you see is exactly what you get.

It is important to bear this in mind, because there are certain operations which can be performed (such as column arithmetic) that work purely with the visual text, and hence accuracy of numbers can be affected if you are not careful in what you are doing.

Also, remember that you can always undo any action you do. This may be especially useful when performing column arithmetic as it can be tricky to get the exact required effect.

Columnar scrap manipulation

CRiSP supports the ability to select rectangular areas from a buffer for scrap manipulation. You can cut, copy and paste columnar regions.

Column Selection

This section describes how to select a column of text in the current buffer so that you can then manipulate it.

There are two main ways to select a column - using the keyboard and the mouse. These are not exclusive. If you are going to select a tall column of text, it can be easier to use the keyboard.

To select an area with the mouse, press the **<Ctrl>** key and use the left mouse button to drag out an area,

To perform the same action using the mouse, press the **<Alt-C>** key to go into column mode. Now as you move the cursor using the arrow keys and other keypad keys, your column will be highlighted. You can unselect a column by pressing **<Alt-C>** again.

If you want to select a column extending to the bottom of the file, you should initially start the column selection using the mouse or keyboard, as described above, and then press **<Ctrl-End>.** This will move the cursor to the last line of the buffer, but maintain the current column position.

Cutting and Copying

You can cut and copy rectangular regions in the same way as non-rectangular regions.

Pasting a column

You need to take care when pasting a column of text. If you use the normal paste operation, then the columnar text will be converted to plain text and you may not achieve the desired result. Instead of the right hand side of the column being aligned, the second and subsequent lines will start at the beginning of the line.

Use this icon, or **Edit ➜ Paste Column** to insert a columnar scrap into the current buffer. If you use the normal paste command, after selecting a columnar region, you will find that CRiSP inserts the text without respecting the columns.

If you use this command, then CRiSP will insert each line of the scrap into the current buffer, but will move down to the next line and insert the next line from the scrap at the same starting column, rather than moving to the beginning of a new line.

The actual effect of this command will differ depending on whether you are in insert mode or overtype mode. If you are in insert mode, then the description above will take place.

If you are in overtype mode, then CRiSP will *overlay* the column on top of the existing text. Effectively, it will delete as many characters from the line as are being inserted. In insert mode, no text deletion is done, and the text already in the buffer slides over to the right to make room for the inserted text.

Column Cut and Paste

Having selected a column, you can perform cut and paste operations with the column, in a similar manner to non-columnar selections.

The one point to be careful with is pasting of columnar data from the scrap. If you use the standard paste commands, then the columnar data will be inserted, but the column shape will not be preserved.

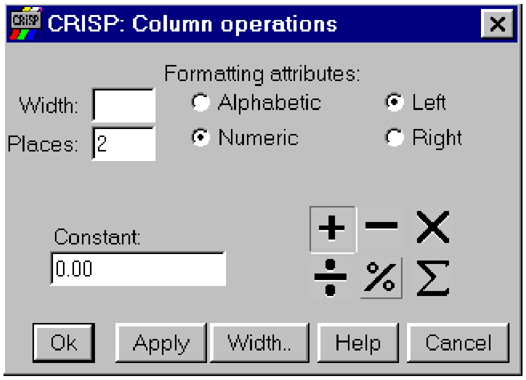
To preserve column shape make sure you use the **Paste Column** menu commands or icons.

Invoked by

Function	**Key**	**Icon**	**Menu**
Copy	**<Keypad-Minus>**	📄	**Edit Copy**
Cut	**<Keypad-Minus>**	✂	**Edit Cut**
Paste	**<Ins>**	📋	**Edit Paste**
Paste column		📋	**Edit Paste column**

Column Operations Dialogue

This dialogue box is used to perform simple formatting or arithmetic on a selected column of data.



Width:

Places: 2

○ Alphabetic

● Numeric

● Left

○ Right

Constant: 0.00

You can use this dialogue box to perform the following actions:

- Left or right justify text.
- Left or right justify numbers.
- Reformat numbers so they have consistent alignment of the decimal point.
- Add, subtract, multiply, divide by a constant all numbers in the selected region.
- Take a percentage of all numbers.
- Sum a column of numbers.

Left and Right Justify Text

When you bring up the dialogue box after selecting a column of text, CRiSP fills in the **Width** field with the width of the column.

If you change the width of a column after bringing up the dialogue box, you should manually change the **Width** field. If you do not do this you may get surprising results.

Select the **Alphabetic** toggle, to set text mode, and select either **Left** or **Right** to select the type of alignment.

You can now click on **Apply** for the command to be executed. In order for anything to happen, there will need to be blanks at either the left edge or right edge of the column. If there are none, then CRiSP will ignore those lines (since they are already justified).

You can pad a field out with extra space by changing the **Width** field to a larger number than the actual size of the field.

You can truncate a field by setting **Width** to a value less than the actual width of the column.

Left and Right Justify Numbers

This action is similar to that described above for text, but CRiSP allows numbers to be reformatted at the same time. Suppose you have a column of figures something like:

```
10
11.5
12
213.10
```

The column of numbers is obviously not aligned as you might expect. Let’s suppose you want to achieve the following layout:

```
 10.00
 11.50
 12.00
213.10
```

I.e. what you want is right justification with 2 decimal places.
This is what you would do on the dialogue box:

Select the **Numeric** option
Select the **Right option**
Set **D.P. Places** to two.
Click on **Apply.**

When CRiSP reformats a column, what it does is to extract the text from the column, reformat the number (using the **D.P. Places** field to force a certain number of digits after the dot). It then puts the result back in to the column but first adding or prefixing enough spaces to match your column **Width**.
If you the width specified and the actual width of the column do not agree, or if the number of decimal places exceeds the column width you may end up with strange results.

Add, Subtract, Multiple, Divide by a Constant

These functions can be performed by selecting one of the arithmetic icons. The default action is to add zero to all numbers. This is fine for reformatting, and is effectively a null operation.
To perform arithmetic on a column, do the following:

Follow the steps given above for reformatting of a column.
Select one of the arithmetic icons.

Specify a constant in the **Constant** field. E.g. if you specify ÷ and a constant of 25, then all numbers will be divided by 25.
Click on **Apply** to do the maths.

Take a percentage of all numbers

Following the steps as above for adding, subtracting, etc, but specify the % icon. The **Constant** should be set to a number as appropriate. (E.g., 100 will leave the numbers as they are; 50 will halve them).

Summing a column of numbers

Use the sigma icon to sum numbers. The actual sum will be displayed at the bottom of the CRiSP window, and the scrap will be replaced by the sum, so that you can paste the value into the current buffer.
An alternative way of doing this is to select the column and execute the **sum** macro at the **Command** line prompt (**<F10>)**..

Notes on arithmetic operations

It is important to bear the following issues in mind when doing number arithmetic. Suppose you have the number **10** in one of your columns, and you are dividing the column by 2, with two decimal places. The result would be **3.33**. You might then multiply the column but instead of getting **10** again, you would get **9.99**. This is because CRiSP does not perform any rounding of the result and any digits which do not fit into the column are dropped and lost. This is different from a spread-sheet, for example, where what you see on screen, and what is stored internally can be different. Usually the internal format of a number has more precision that what you see on screen.

Column Width

The column width dialogue box is a convenient way to pad out or trim columns of text (or numbers). Columns with numbers in them are treated exactly the same way as columns of text.
You can invoke this dialogue box by selecting the **Width...** button from the **Column Operations** dialogue box (**Edit Columns**).
You are provided with a number of buttons. In order to make effective use of this dialogue box, select the column of text before using the buttons:

Widening a column

<<Tab This option will add a Tab character to the left side of the column.
<-Space will add a single space to the left side of the column.
Space-> will add a single space to the right side of the column.

Tab>> This option will add a Tab character to the right side of the column.

Slimming a column

Slim left This will remove a single space at the left side of a column.
Slim right This will remove a single space at the right side of a column.

Further Information

You can obtain further information on using CRiSP by consulting the on-line guides, which are hypertext based (allow you to quickly jump between related sections), and contains a complete index allowing you to search for topics of interest.
Updated copies of the software and platform availability information is available from your software supplier or on the Internet at this address:

http://www.pacemaker.co.uk

Troubleshooting License Problems

First recheck that you have followed all instructions on the password sheet. If you still have problems then check the following:

License Problems (UNIX)

If CRiSP fails to get a license check the following:

- The license is identical to the hard copy.
- Check to see if the license is valid:-

  lmcrisp -v <license-file>
  The license file will need to be specified if there is a license manager already running.
- If the license does **<u>not</u>** check out OK then contact your distributor.
- If the license checks out OK, but CRiSP still fails to get a license, check to see if the lmcrisp process is running.

If you change the license file in any way, before starting CRiSP you will need to kill of the lmcrisp process and restart the license manager.

- Check you have set the DDMF_HOST as explained on the password sheet.

License Problems (Windows 95)

If CRiSP fails to get a license check the following:

- The license is identical to the hard copy.
- If you have used the Notepad editor to create your license.dat file, make sure that it is really called that by looking at the file through DOS. You may find that the Window’s 95 file manager calls the file license.dat, but in reality it has been called license.dat.txt which CRiSP doesn’t recognise.

Index