Reference Manual

Volume III
Interfaces Guide

Quicksilver Beta

March 22nd 2008

CLIPS Interfaces Guide
Quicksilver Beta   March 22nd 2008

CONTENTS

License Information

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

CLIPS is released as public domain software and as such you are under no obligation to pay for its use. However, if you derive commercial or monetary benefit from use of the software or just want to show support, please consider making a voluntary payment based on the worth of the software to you as compensation for the time and effort required to develop and maintain CLIPS. Payments can be made online at http://order.kagi.com/?JKT.

Preface

The History of CLIPS

The origins of the C Language Integrated Production System (CLIPS) date back to 1984 at NASA's Johnson Space Center. At this time, the Artificial Intelligence Section (now the Software Technology Branch) had developed over a dozen prototype expert systems applications using state-of-the-art hardware and software. However, despite extensive demonstrations of the potential of expert systems, few of these applications were put into regular use. This failure to provide expert systems technology within NASA's operational computing constraints could largely be traced to the use of LISP as the base language for nearly all expert system software tools at that time. In particular, three problems hindered the use of LISP based expert system tools within NASA: the low availability of LISP on a wide variety of conventional computers, the high cost of state-of-the-art LISP tools and hardware, and the poor integration of LISP with other languages (making embedded applications difficult).

The Artificial Intelligence Section felt that the use of a conventional language, such as C, would eliminate most of these problems, and initially looked to the expert system tool vendors to provide an expert system tool written using a conventional language. Although a number of tool vendors started converting their tools to run in C, the cost of each tool was still very high, most were restricted to a small variety of computers, and the projected availability times were discouraging. To meet all of its needs in a timely and cost effective manner, it became evident that the Artificial Intelligence Section would have to develop its own C based expert system tool.

The prototype version of CLIPS was developed in the spring of 1985 in a little over two months. Particular attention was given to making the tool compatible with expert systems under development at that time by the Artificial Intelligence Section. Thus, the syntax of CLIPS was made to very closely resemble the syntax of a subset of the ART expert system tool developed by Inference Corporation. Although originally modelled from ART, CLIPS was developed entirely without assistance from Inference or access to the ART source code.

The original intent for CLIPS was to gain useful insight and knowledge about the construction of expert system tools and to lay the groundwork for the construction of a replacement tool for the commercial tools currently being used. Version 1.0 demonstrated the feasibility of the project concept. After additional development, it became apparent that CLIPS would be a low cost expert system tool ideal for the purposes of training. Another year of development and internal use went into CLIPS improving its portability, performance, functionality, and supporting documentation. Version 3.0 of CLIPS was made available to groups outside of NASA in the summer of 1986.

Further enhancements transformed CLIPS from a training tool into a tool useful for the development and delivery of expert systems as well. Versions 4.0 and 4.1 of CLIPS, released

respectively in the summer and fall of 1987, featured greatly improved performance, external language integration, and delivery capabilities. Version 4.2 of CLIPS, released in the summer of 1988, was a complete rewrite of CLIPS for code modularity. Also included with this release were an architecture manual providing a detailed description of the CLIPS software architecture and a utility program for aiding in the verification and validation of rule-based programs. Version 4.3 of CLIPS, released in the summer of 1989, added still more functionality.

Originally, the primary representation methodology in CLIPS was a forward chaining rule language based on the Rete algorithm (hence the Production System part of the CLIPS acronym). Version 5.0 of CLIPS, released in the spring of 1991, introduced two new programming paradigms: procedural programming (as found in languages such as C and Ada) and object-oriented programming (as found in languages such as the Common Lisp Object System and Smalltalk). The object-oriented programming language provided within CLIPS is called the CLIPS Object-Oriented Language (COOL). Version 5.1 of CLIPS, released in the fall of 1991, was primarily a software maintenance upgrade required to support the newly developed and/or enhanced X Window, MS-DOS, and Macintosh interfaces. Version 6.0 of CLIPS, released in 1993, provided support for the development of modular programs and tight integration between the object-oriented and rule-based programming capabilities of CLIPS. Version 6.1 of CLIPS, released in 1998, removed support for older non-ANSI C Compilers and added support for C++ compilers. Commands to profile the time spent in constructs and user-defined functions were also added. Version 6.2 of CLIPS, released in 2002, added support for multiple environments into which programs can be loaded and improved Windows XP and MacOS development interfaces.

Because of its portability, extensibility, capabilities, and low cost, CLIPS has received widespread acceptance throughout the government, industry, and academia. The development of CLIPS has helped to improve the ability to deliver expert system technology throughout the public and private sectors for a wide range of applications and diverse computing environments. CLIPS is being used by numerous users throughout the public and private community including: all NASA sites and branches of the military, numerous federal bureaus, government contractors, universities, and many private companies.

CLIPS is now maintained as public domain software by the main program authors who no longer work for NASA. See appendix A of the *Basic Programming Guide* for information on obtaining CLIPS and support.

CLIPS Version 6.3

Version 6.3 of CLIPS contains two major enhancements. First, rule performance has been improved particularly in situations with large numbers of facts/instances or partial matches. Second, support for integration with languages such as Java and C++ has been improved. For a detailed listing of differences between the 6.x releases of CLIPS, refer to appendix B of the *Basic Programming Guide* and appendix C of the *Advanced Programming Guide*.

CLIPS Documentation

Two documents are provided with CLIPS.

- The *CLIPS Reference Manual* which is split into the following parts:

  - *Volume I - The Basic Programming Guide*, which provides the definitive description of CLIPS syntax and examples of usage.

  - *Volume II - The Advanced Programming Guide*, which provides detailed discussions of the more sophisticated features in CLIPS and is intended for people with extensive programming experience who are using CLIPS for advanced applications.

  - *Volume III - The Interfaces Guide*, which provides information on machine-specific interfaces.

- The *CLIPS User's Guide* which provides an introduction to CLIPS rule-based and object-oriented programming and is intended for people with little or no expert system experience.

Acknowledgements

As with any large project, CLIPS is the result of the efforts of numerous people. The primary contributors have been: Robert Savely, who conceived the project and provided overall direction and support; Chris Culbert, who managed the project and wrote the original *CLIPS Reference Manual*; Gary Riley, who designed and developed the rule-based portion of CLIPS, co-authored the *CLIPS Reference Manual*, and developed the Macintosh interface for CLIPS; Brian Dantes, who designed and developed the CLIPS Object Oriented Language (COOL) and co-authored the *CLIPS Reference Manual*; Bebe Ly, who developed the X Window interface for CLIPS; Chris Ortiz, who developed the original Windows 95 interface for CLIPS; Dr. Joseph Giarratano of the University of Houston-Clear Lake, who wrote the *CLIPS User's Guide*; and Frank Lopez, who designed and developed CLIPS version 1.0 and wrote the CLIPS 1.0 User's Guide.

Many other individuals contributed to the design, development, review, and general support of CLIPS, including: Jack Aldridge, Carla Colangelo, Paul Baffes, Ann Baker, Stephen Baudendistel, Les Berke, Tom Blinn, Marlon Boarnet, Dan Bochsler, Bob Brown, Barry Cameron, Tim Cleghorn, Major Paul Condit, Major Steve Cross, Andy Cunningham, Dan Danley, Mark Engelberg, Kirt Fields, Ken Freeman, Kevin Greiner, Ervin Grice, Sharon Hecht, Patti Herrick, Mark Hoffman, Grace Hua, Gordon Johnson, Phillip Johnston, Sam Juliano, Ed Lineberry, Bowen Loftin, Linda Martin, Daniel McCoy, Terry McGregor, Becky McGuire, Scott Meadows, C. J. Melebeck, Paul Mitchell, Steve Mueller, Bill Paseman, Cynthia Rathjen, Eric Raymond, Reza Razavipour, Marsha Renals, Monica Rua, Tim Saito, Michael Sullivan, Gregg Swietek, Eric Taylor, James Villarreal, Lui Wang, Bob Way, Jim Wescott, Charlie Wheeler, and Wes White.

Section 1 - Introduction

This manual is the *Interfaces Guide* for CLIPS. It is intended for users interested in using the machine specific interfaces for CLIPS. Section 2 of this manual describes the IBM PC compatible Windows interface for CLIPS, Section 3 describes the Macintosh interface for CLIPS, and Section 4 describes the X Window interface for CLIPS.

Section 2 - CLIPS Windows Interface

This section provides a brief summary of each menu command found in the CLIPS 6.3 Windows interface. The menus are listed in the left to right order in which they appear in the menu bar. The commands within each menu are also listed in the order in which they appear in the menu.

The rest of the CLIPS 6.3 Windows interface is straightforward, following Windows interface standards. Commands can be entered directly in the dialog window in a fashion similar to the standard CLIPS command line interface. The integrated editor is implemented in a fashion similar to that of other Windows editors.

Inline editing is supported in the dialog window. The left and right arrow keys can be used to move the caret backwards and forwards through the current command. Holding the shift key down when the left or right arrow is pressed moves the caret respectively to the next line beginning or next line end of the current command. Pressing the delete key will delete the character to the left of the caret. Insertion of other characters or pasted text occurs at the caret. The esc key moves the caret to the end of the current command. The caret must be at the end of the current command in order for pressing the return key to execute the command.

A command history is also supported for the dialog window. The up and down arrows allow you to cycle through the command history. The up arrow restores the previous command and the down arrow restores the next command. Holding the shift key down when the up or down arrow is pressed takes you respectively to the beginning or end of the command history.

From the CLIPS command prompt, the command **clear-window** (which takes no arguments) will also clear all of the text in the dialog window.

If rules are not being executed, then execution of the current command or CLIPS program can be halted by holding down the **control** key while pressing the period key. If rules are being executed, then rule execution can be interrupted on a rule firing boundary by holding the **control** key while pressing the period key. Holding down the shift key, the **control** key, and the period key will halt rule execution after the current RHS action. The **Halt** menu item can also be selected from the **Execution** menu during execution. Selecting this menu item is equivalent to holding down the **control** key while pressing the period key. If the shift key is held down before clicking in the menu bar, the **Halt** menu item is changed to the **Halt Now!** menu item. Selecting this menu item is equivalent to holding down the shift key and the **control** key while pressing the period key.

Some menu commands are available while CLIPS is executing a user program. Among these commands are the **Options...** menu item, the **Watch...** menu item, the **Turn Dribble On.../Turn Dribble Off** menu item, and all menu items in **Window** menu. The command summary for each menu item indicates whether it is available while CLIPS is executing. In addition, windows may be moved and re-sized while CLIPS is executing.

2.1 THE FILE MENU

File	
New	Ctrl+N
Open...	Ctrl+O
Load...	Ctrl+L
Load Batch...	
Load Binary...	
Turn Dribble On...	
Close	
Save	Ctrl+S
Save As...	
Save Binary...	
Revert	
Page Setup...	
Print...	Ctrl+P
Exit	
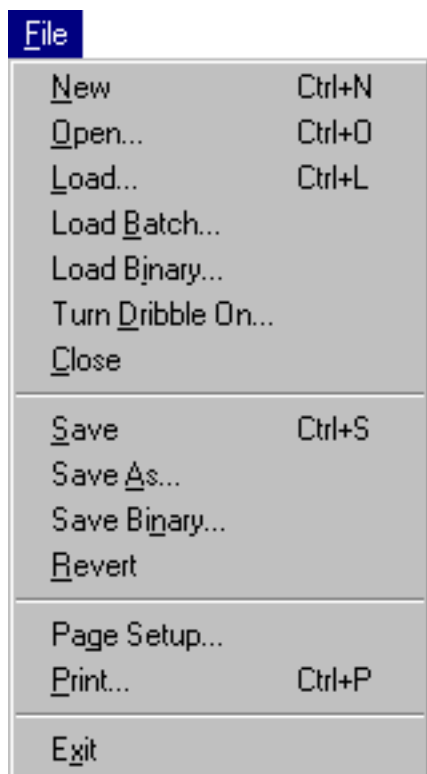
2.1.1 New (Ctrl+N)

This command opens a new buffer for editing with the window name Untitled.

2.1.2 Open... (Ctrl+O)

This command displays the standard Windows file selection dialog box, allowing the user to select a text file to be opened as a buffer for editing.

2.1.3 Load... (Ctrl+L)

This command displays the standard Windows file selection dialog box, allowing the user to select a text file to be loaded into the knowledge base. This command is equivalent to the CLIPS command (load <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS load command will be echoed to the dialog window and executed. This command is not available when CLIPS is executing.

Note: when using the CLIPS command (load <file-name>) specify the path using the forward slash character '/' or a double backslash '\\' instead of the standard DOS delimiter '\'.

For example,

```
(load "C: / CLIPSWIN / EXAMPLES / TEST.CLP")
```

or

```
(load "C: \\ CLIPSWIN \\ EXAMPLES \\ TEST.CLP")
```

2.1.4 Load Batch...

This command displays the standard Windows file selection dialog box, allowing the user to select a text file to be executed as a batch file. This command is equivalent to the CLIPS command (batch <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS batch command will be echoed to the dialog window and executed. This command is not available when CLIPS is executing.

2.1.5 Load Binary...

This command displays the standard Windows file selection dialog box, allowing the user to select a file to be loaded as a binary image. This command is equivalent to the CLIPS command (bload <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS bload command will be echoed to the dialog window and executed. This command is not available when CLIPS is executing.

2.1.6 Turn Dribble On...

This command displays the standard Windows file selection dialog box, allowing the user to select a text file in which subsequent display window output will be stored. This command is equivalent to the CLIPS command (dribble-on <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS dribble-on command will be echoed to the dialog window and executed. While the dribble file is active, this menu item will be modified to read **Turn Dribble Off**. Selecting the menu item at this time will close the dribble text file. The **Turn Dribble Off** command is equivalent to the CLIPS command (dribble-off). This command is available when CLIPS is executing (however the commands will not be echoed to the dialog window).

2.1.7 Close

This command closes the currently active edit or status window.

2.1.8 Save (Ctrl+S)

This command saves the file in the active edit window or the contents of the dialog or status window. If the file is Untitled, the Dialog window, or a status window , a dialog box will prompt for a file name under which to save the file. The **file type** option in the standard file dialog box determines whether a file is saved as a batch file or a text file. CLIPS batch file documents are executed as a series of commands when launched, whereas CLIPS text file documents are placed in an editing buffer when launched. Both types of documents can be edited as text files in the editor.

2.1.9 Save As...

This command allows the active edit window to be saved under a new name. A dialog box will appear to prompt for the new file name. The name of the editing window will be changed to the new file name.

2.1.10 Save Binary...

This command allows the constructs currently stored in CLIPS to be saved as a binary image. A dialog box will appear to prompt for the new file name in which to store the binary image. This command is equivalent to the CLIPS command (bsave <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS bsave command will be echoed to the dialog window and executed. This command is not available when CLIPS is executing.

2.1.11 Revert

This command restores the active edit window to the last-saved version of the file in the buffer. Any changes made since the file was last saved will be discarded.

2.1.12 Page Setup...

This command allows the user to specify information about the size of paper used by the printer.

2.1.13 Print... (Ctrl+P)

This command allows the user to print the active edit window.

2.1.14 Exit

This command exits CLIPS. This command is not available when CLIPS is executing.

2.2 THE EDIT MENU
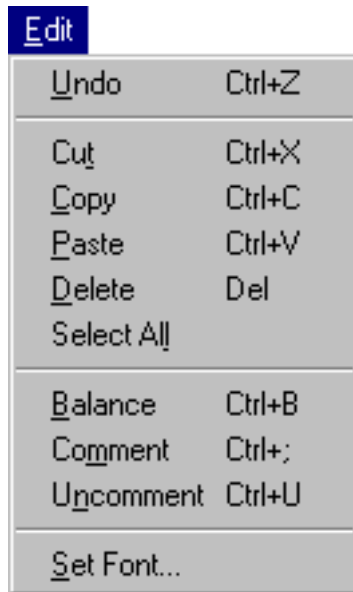
2.2.1 Undo (Ctrl+Z)

This command allows you to undo your last editing operation. Typing, cut, copy, paste, and delete operations can all be undone.

2.2.2 Cut (Ctrl+X)

This command removes selected text in the edit window and places it in the Clipboard.

2.2.3 Copy (Ctrl+C)

This command copies selected text in the edit window and places it in the Clipboard.

2.2.4 Paste (Ctrl+V)

This command copies the contents of the Clipboard to the selection point in the edit window or the Dialog window. If there is selected text in the edit window, it is replaced by the contents of the Clipboard.

2.2.5 Delete

This command removes selected text in the edit window. The selected text is not placed in the Clipboard.

2.2.6 Select All

This command selects all of the text in the active edit window.

2.2.7 Balance (Ctrl+B)

This command operates on the current selection in the edit window by attempting to find the smallest selection containing the current selection which has balanced parentheses. Repeatedly using this command will select larger and larger selections of text until a balanced selection cannot be found. The balance command is a purely textual operation and can be confused by parentheses found in strings.

2.2.8 Comment (Ctrl+;)

This command operates on the current selection in the active edit window by adding a semicolon to the beginning of each line contained in the selection.

2.2.9 Uncomment (Ctrl+U)

This command operates on the current selection in the active edit window by removing a semicolon (if one exists) from the beginning of each line contained in the selection.

2.2.10 Set Font...

This command allows the font used in the currently active editing buffer to be changed. When you use this command, the following dialog will appear:
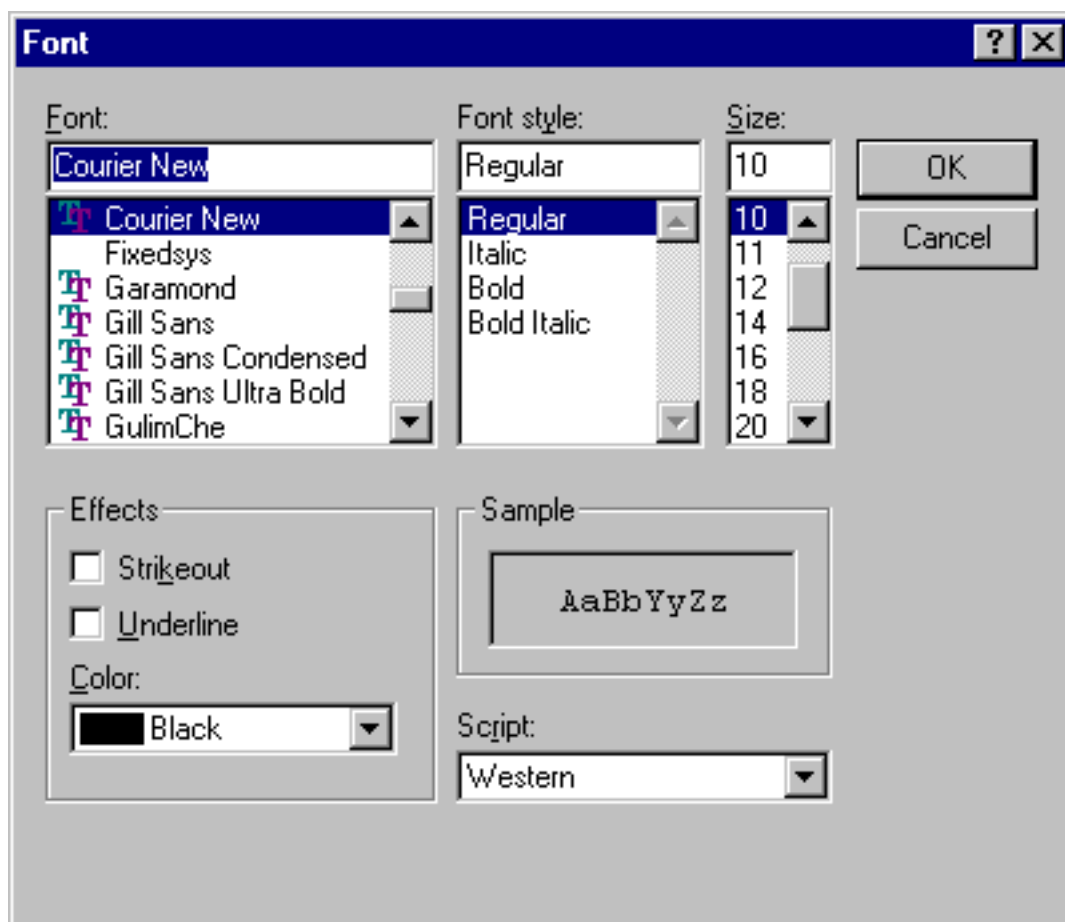
Select the desired font and font attributes in the dialog. The letters in the Sample area will be displayed using the selected font. Click the **OK** button to change the font used in the currently active editing buffer. Click the **Cancel** button to retain the current font setting.

2.3 THE BUFFER MENU
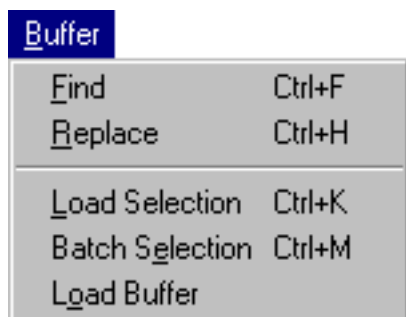
2.3.1 Find (Ctrl+F)

This command displays a dialog box which allows the user to set parameters for text search operations. The dialog box that appears allows a search string to be specified.

The **Match Case** option makes the string search operation case-insensitive for alphabetic characters; that is, the string "Upper" will not match the string "uPPER". The default for this setting is off.

Once search options have been set, one of three search dialog buttons can be pressed. The **Find Next** button will search for teach occurrence of the string in the edit window starting at the end of the current selection and preceding to the end of the buffer. The **Cancel** button aborts the find command. The search options are restored to the values they had before the dialog box was entered.

2.3.2 Replace (Ctrl+H)

This command displays a dialog box which allows the user to set parameters for text search and replacement operations. The dialog box that appears allows a search and replacement string to be specified.
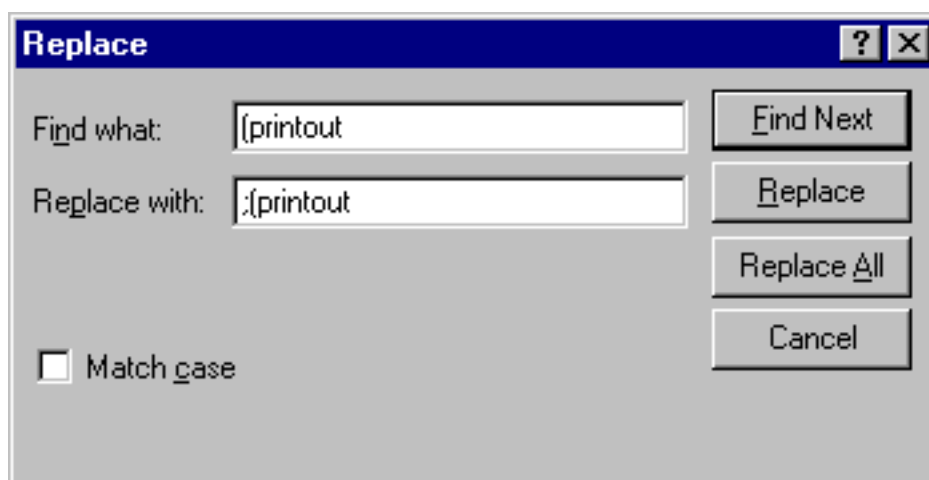
The **Match Case** option makes the string search operation case-insensitive for alphabetic characters; that is, the string "Upper" will not match the string "uPPER". The default for this setting is off.

Once search options have been set, one of three search dialog buttons can be pressed. The **Find Next** button will search for each occurrence of the "**Find What**" string in the edit window starting at the end of the current selection and preceding to the end of the buffer. The **Replace** button will replace one instance of the "**Find What**" string with the "**Replace With**" string. The **Replace All** button will replace every instance of the "**Find What**" string with the "**Replace With**" string without user verification. The **Cancel** button aborts the replace command. The search and replace strings and options are restored to the values they had before the dialog box was entered.

2.3.3 Load Selection (Ctrl+K)

This command loads the current selection from the edit window into the CLIPS knowledge base. Standard error detection and recovery routines used to load constructs from a file are also used when loading a selection (i.e., if a construct has an error in it, the rest of the construct will be skipped over until another construct to be loaded is found). This command is not available when CLIPS is executing.

2.3.4 Batch Selection (Ctrl+M)

This command treats the current selection in the edit window as if it were a batch file and executes it as a series of commands. Standard error detection and recovery routines used to load construct from a file are *not* used when batching a selection (i.e., if a construct has an error in it, a number of ancillary errors may be generated by subsequent parts of the same construct following the error). This command is not available when CLIPS is executing.

2.3.5 Load Buffer

This command loads the contents of the active edit window into the CLIPS knowledge base. It is equivalent to selecting the entire buffer and executing a **Load Selection** command. This command is not available when CLIPS is executing.

2.4 THE EXECUTION MENU
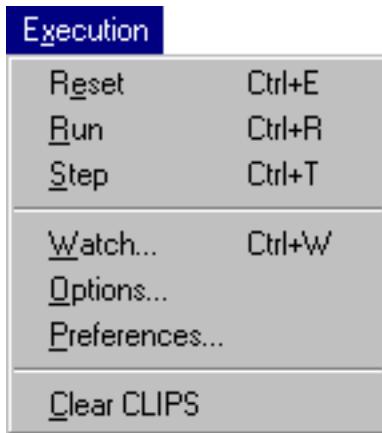
2.4.1 Reset (Ctrl+E)

This command is equivalent to the CLIPS command (reset). When this command is chosen, the CLIPS command (reset) will be echoed to the dialog window and executed. If Warnings are enabled and activation's are currently on the agenda, a dialog box will appear issuing a warning and providing an opportunity to cancel this command. This command is not available when CLIPS is executing.

2.4.2 Run (Ctrl+R)

This command is equivalent to the CLIPS command (run). When this command is chosen, the CLIPS command (run) will be echoed to the dialog window and executed. During execution, this menu command is changed to **Halt** and can be selected to halt execution (as if a command period had been entered). Holding the shift key down while selecting this menu item is equivalent to entering command-shift-period. For an explanation of the different methods of halting execution refer to the introduction at the beginning of this section.

2.4.3 Step (Ctrl+T)

This command is equivalent to the CLIPS command (run <limit>) where <limit> is the value for the **Step Rule Firing Increment**. (it is set using the **Options...** command under the Edit menu). When this command is chosen, the CLIPS command (run <limit>) will be echoed to the dialog window and executed. If the default value for the **Step Rule Firing Increment** is used, this command is equivalent to the CLIPS command (run 1). This command is not available when CLIPS is executing.

2.4.4 Watch... (Ctrl+W)

This command displays a dialog box which allows the user to set various watch items as enabled or disabled.

Watch items can be enabled or disabled by clicking the appropriate check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item check boxes. Pressing the **None** button unchecks all of the watch item check boxes. Clicking the **OK** button exits the dialog and changes the current watch settings to those shown in the dialog. Clicking the **Cancel** button exits the dialog, but retains the original settings of the watch items before the dialog was entered.

This command is available when CLIPS is executing.

2.4.5 Options...

This command displays a dialog box which allows the user to set various CLIPS execution options.

The **Salience Evaluation** pop-up menu allows the current salience evaluation behavior to be changed (to either when-defined, when-activated, or every-cycle). The **Strategy** pop-up menu allows the current conflict resolution strategy to be changed (to either depth, breadth, lex, mea, complexity, simplicity, or random). The **Static Constraint Checking**, **Dynamic Constraint Checking**, **Reset Global Variables**, **Sequence Expansion Operator Recognition**, **Incremental Reset**, **Auto-Float Dividend**, and **Fact Duplication** check boxes are used to enable and disable the named CLIPS option. Enabled options have a check in their check box while disabled items have none. The **Incremental Reset** option cannot be changed if any defrules are currently defined. Clicking the **OK** button exits the dialog and changes the CLIPS execution options settings to those shown in the dialog. Clicking the **Cancel** button exits the dialog, but retains the original settings of the CLIPS execution options before the dialog was entered.

This command is available when CLIPS is executing.

2.4.6 Preferences...

This command displays a dialog box which allows the user to set the parameters for several options in the CLIPS Windows interface.

Enable Warnings option determines whether certain warnings are issued. The default for this option is on. When Warnings are enabled, certain commands chosen from the menu bar will display a dialog box giving the user the option to abort the command. For example, executing a clear command with Warnings enabled will display the following dialog box:
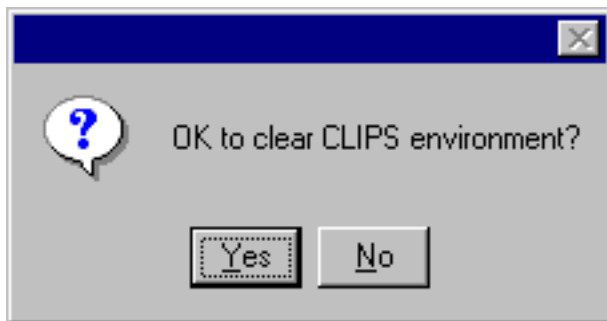
Warnings will only be issued for commands chosen from the menu bar. For example, typing a clear command directly into the dialog window and executing it will not display a warning dialog box regardless of whether Warnings are enabled.

The **Command Completion Dialog** option determines whether a dialog is displayed to determine which completion to use if more than one completion is available. The default for this option is on. That is, a dialog will be displayed whenever there is more than one possible completion. If this option is off, then a beep is sounded when no possible completion's exist. From the CLIPS command prompt, the command **set-completion-dialog** can be used to enable and disable the use of the completion dialog when there is more than one choice (pass the symbol *TRUE* to enable the completion dialog and the symbol *FALSE* to disable it).

The **Step Rule Firing Increment** option determines how many rules will be executed when the **Step** command (see Execution menu) is issued. Any integer between 1 and 999 inclusive can be entered for this option. The default value for this option is 1.

 This command is available when CLIPS is executing.

2.4.7 Clear CLIPS

This command is equivalent to the CLIPS command (clear). When this command is chosen, the CLIPS command (clear) will be echoed to the dialog window and executed. If Warnings are enabled, a dialog box will appear issuing a warning and providing an opportunity to cancel this command. This command is not available when CLIPS is executing.

2.5 THE BROWSE MENU
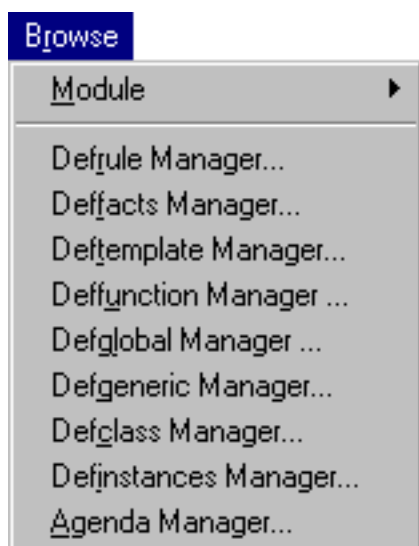
2.5.1 The Module Menu
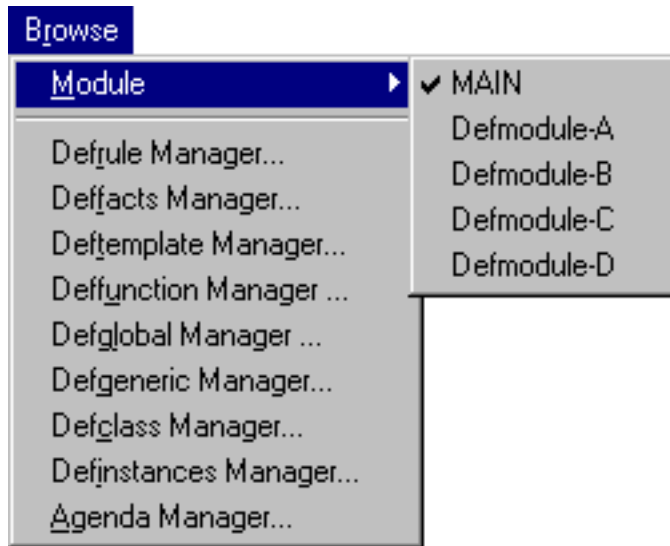
This menu displays a list of the defmodules currently defined in the CLIPS environment. A check mark is displayed by the current module. Selecting a module from the menu list changes the current module to the selected item.

2.5.2 Defrule Manager...

This command displays a dialog box which allows the defrules in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defrule selected from the list of defrules currently in the knowledge base.

The **Remove** button will remove the currently selected defrule. Pressing this button is equivalent to issuing an **undefrule** command. The **Refresh** button will refresh the currently selected defrule. Pressing this button is equivalent to a **refresh** command. The **Matches** command will list the matches for the currently selected defrule. Pressing this button is equivalent to a **matches** command. The **Pprint** button will pretty print the currently selected defrule. Pressing this button is equivalent to a **ppdefrule** command. Both the **Matches** and **Pprint** buttons echo the equivalent CLIPS command and resulting output to the dialog window.

The **Breakpoint** check box is used to enable or disable a breakpoint on the currently selected defrule. Rules with a breakpoint set have a check in their check box while rules without a breakpoint have none. The **Watch Activations** check box is used to enable or disable the messages that are printed when a rule is activated or deactivated. If the check box for a rule is checked, then activation/deactivation messages for the rule will be printed. The **Watch Firings** check box is used to enable or disable the messages that are printed when a rule is executed. If the check box for a rule is checked, then execution messages for the rule will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defrule Manager...** command is not available when CLIPS is executing.

2.5.3 Deffacts Manager...

This command displays a dialog box which allows the deffacts in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffacts selected from the list of deffacts currently in the knowledge base.

The **Remove** button will remove the currently selected deffacts. Pressing this button is equivalent to an **undeffacts** command The **Pprint** button will pretty print the currently selected deffacts. Pressing this button is equivalent to a **ppdeffacts** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deffacts Manager...** command is not available when CLIPS is executing.

2.5.4 Deftemplate Manager...

This command displays a dialog box which allows the deftemplates in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deftemplate selected from the list of deftemplates currently in the knowledge base.

The **Remove** button will remove the currently selected deftemplate. Pressing this button is equivalent to an **undeftemplate** command. The **Pprint** button will pretty print the currently selected deftemplate. Pressing this button is equivalent to a **ppdeftemplate** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed whenever a fact is asserted or retracted. If the check box for a deftemplate is checked, then assert/retract messages for facts associated with the deftemplate will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deftemplate Manager...** command is not available when CLIPS is executing.

2.5.5 Deffunction Manager...

This command displays a dialog box which allows the deffunctions in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffunction selected from the list of deffunctions currently in the knowledge base.

The **Remove** button will remove the currently selected deffunction. Pressing this button is equivalent to an **undeffunction** command. The **Pprint** button will pretty print the currently selected deffunction. Pressing this button is equivalent to a **ppdeffunction** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a deffunction begins and ends execution. If the check box for a deffunction is checked, then execution messages for the deffunction will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deffunction Manager...** command is not available when CLIPS is executing.

2.5.6 Defglobals Manager...

This command displays a dialog box which allows the defglobals in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defglobal selected from the list of defglobals currently in the knowledge base.

The **Remove** button will remove the currently selected defglobal. Pressing this button is equivalent to an **undefglobal** command. The **Pprint** button will pretty print the currently selected defglobal. Pressing this button is equivalent to a **ppdefglobal** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defglobals Manager...** command is not available when CLIPS is executing.

2.5.7 Defgeneric Manager...

This command displays a dialog box which allows the defgenerics in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defgeneric selected from the list of defgenerics currently in the knowledge base. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defgeneric Manager...** command is not available when CLIPS is executing.

The **Remove** button will remove the currently selected defgeneric. Pressing this button is equivalent to an **undefgeneric** command. The **Pprint** button will pretty print the currently selected defgeneric. Pressing this button is equivalent to a **ppdefgeneric** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a generic function begins and ends execution. If the check box for a deffunction is checked, then execution messages for the generic function will be printed.

The **Methods...** button displays a dialog box allowing the defmethods for the currently selected defgeneric to be browsed. The dialog box that appears in response to this button allows an operation to be performed on a defmethod selected from the list of defmethods currently in the knowledge base.

The **Remove** button will remove the currently selected defmethod. Pressing this button is equivalent to an **undefmethod** command. The **Pprint** button will pretty print the currently selected defmethod. Pressing this button is equivalent to a **ppdefmethod** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a specific method of a generic function begins and ends execution. If the check box for a defmethod is checked, then execution messages for the method will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to return control to the **Defgeneric Manager** dialog box.

2.5.8 Defclass Manager...

This command displays a dialog box which allows the defclasses in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defclass selected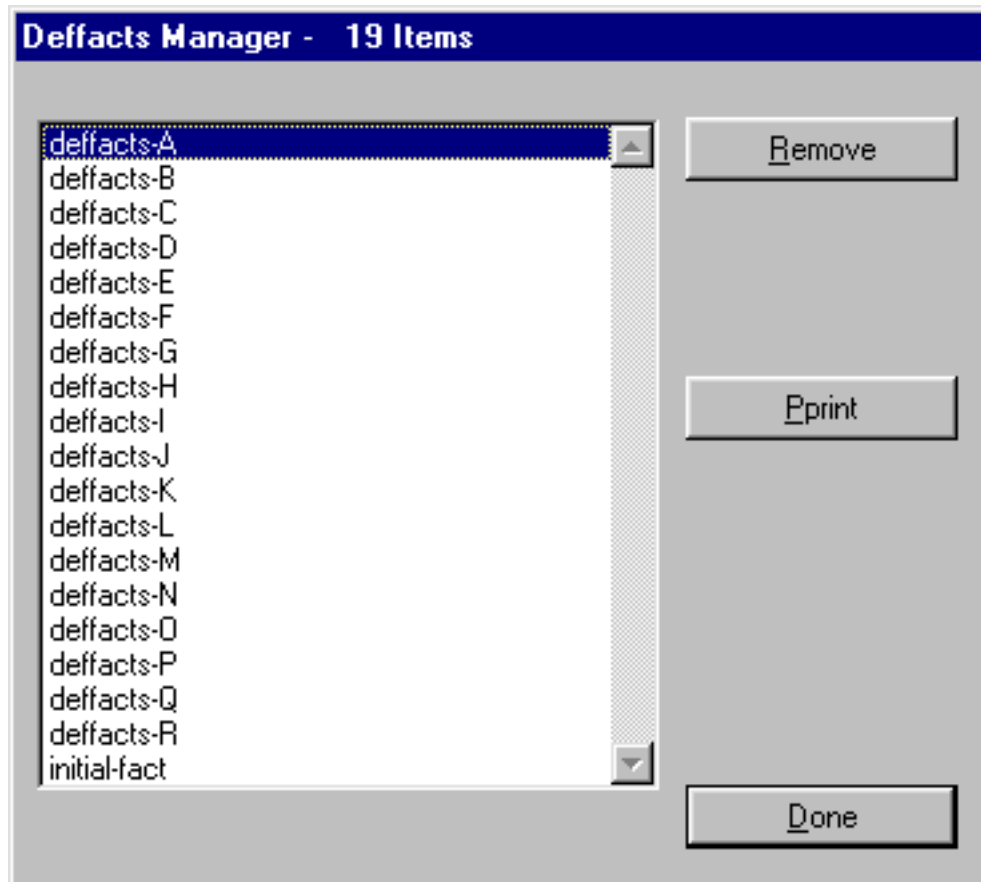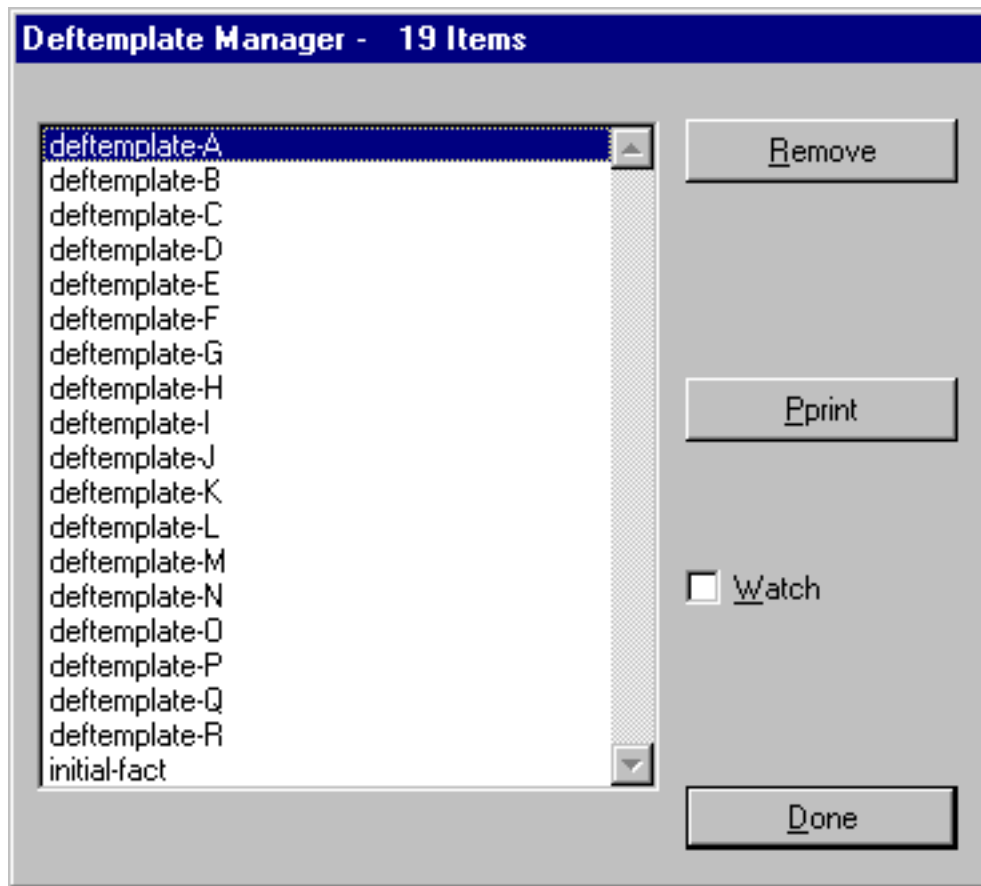 from the list of defclasses currently in the knowledge base. Any number of operations can be executed from this dialog. When finished, click the **Done** button to

remove the dialog. The **Defclass Manager...** command is not available when CLIPS is executing.

The **Remove** button will remove the currently selected defclass. Pressing this button is equivalent to an **undefclass** command. The **Describe** button will describe the currently selected defclass. Pressing this button is equivalent to a **describe-class** command. The **Browse** command displays the class hierarchy for the currently selected defclass. Pressing this button is equivalent to a **browse-class** command. The **Pprint** button will pretty print the currently selected defclass. Pressing this button is equivalent to a **ppdefclass** command. The **Describe**, **Browse**, and **Pprint** buttons echo the equivalent CLIPS command and resulting output to the dialog window.

The **Watch Instances** check box is used to enable or disable the messages that are printed when an instance is created or deleted. If the check box for a defclass is checked, then messages will be printed whenever an instance of that class is created or deleted. The **Watch Slots** check box is used to enable or disable the messages that are printed when an instance's slot values are changed. If the check box for a defclass is checked, then messages will be printed whenever an instance of that class has a slot value changed.

The **Message Handlers...** button displays a dialog box allowing the message handlers for the currently selected class to be browsed. The dialog box that appears in response to this button

allows an operation to be performed on a defmessage-handler selected from the list of defmessage-handlers currently in the knowledge base.

The **Remove** button will remove the currently selected defmessage-handler. Pressing this button is equivalent to an **undefmessage-handler** command. The **Pprint** button will pretty print the currently selected defmessage-handler. Pressing this button is equivalent to a **ppdefmessage-handler** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a message-handler begins and ends execution. If the check box for a defmessage-handler is checked, then execution messages for the message-handler will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to return control to the **Defclass Manager** dialog box.

2.5.9 Definstances Manager...

This command displays a dialog box which allows the definstances in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a definstances selected from the list of definstances currently in the knowledge base.
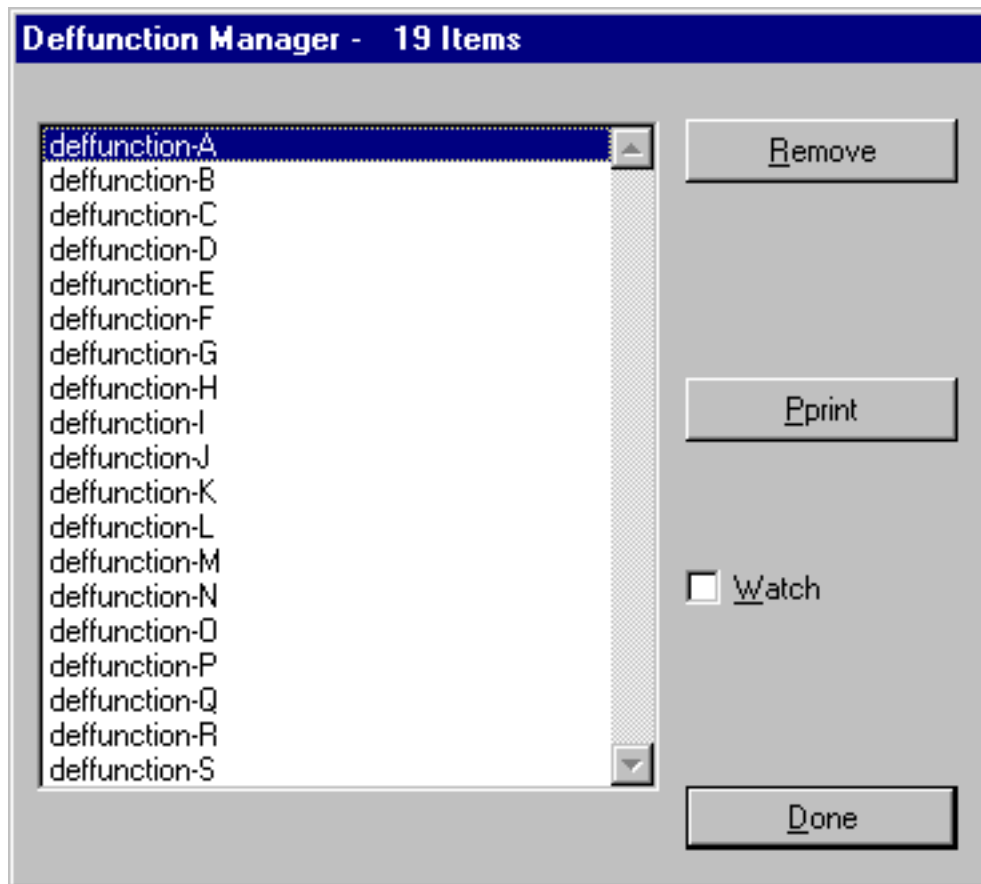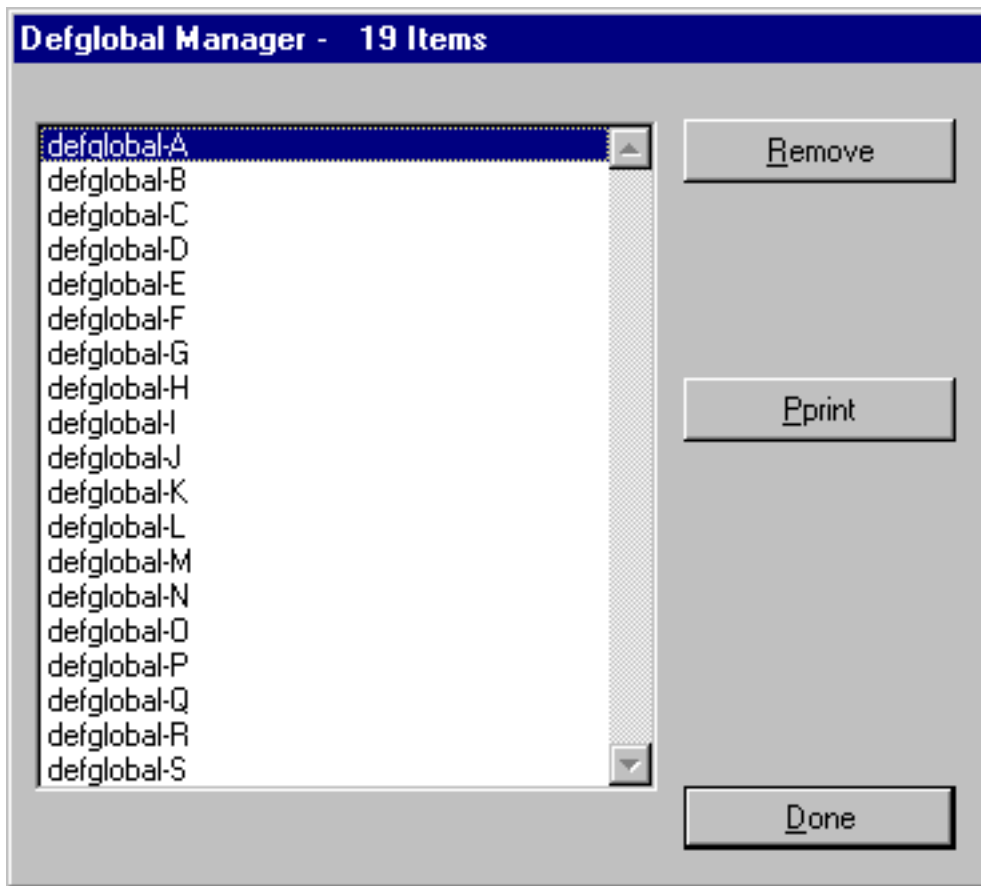

The **Remove** button will remove the currently selected definstances. Pressing this button is equivalent to an **undefinstances** command. The **Pprint** button will pretty print the currently selected definstances. Pressing this button is equivalent to a **ppdefinstances** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Definstances Manager...** command is not available when CLIPS is executing.

2.5.10 Agenda Manager...

This command displays a dialog box which allows the activation's on the agenda to be examined. The dialog box that appears in response to this command allows any activation on the agenda to be removed or fired.

```
Agenda Manager -   19 Items

0   defrule-3: f-0,f-7                    Remove
0   defrule-3: f-2,f-7
0   defrule-3: f-3,f-7
0   defrule-3: f-5,f-7
0   defrule-1: f-5,f-6
0   defrule-3: f-2,f-6
0   defrule-3: f-5,f-6
0   defrule-3: f-5,f-4
0   defrule-2: f-5
0   defrule-3: f-0,f-4                      Fire
0   defrule-3: f-2,f-4
0   defrule-3: f-3,f-4
0   defrule-3: f-3,f-1
0   defrule-2: f-3
0   defrule-1: f-2,f-1
0   defrule-3: f-2,f-1
0   defrule-2: f-2
0   defrule-3: f-0,f-1
0   defrule-2: f-0                          Done
```

The **Remove** button will remove the currently selected activation from the agenda. The **Fire** button will place the currently selected activation at the top of the agenda (regardless of its salience). The dialog is then exited and the CLIPS command (run 1) will then be echoed to the **Dialog Window**, causing the activation to fire. Any number of remove operations (except the **Fire** button) can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Agenda Manager...** command is not available when CLIPS is executing.

2.6 THE WINDOW MENU
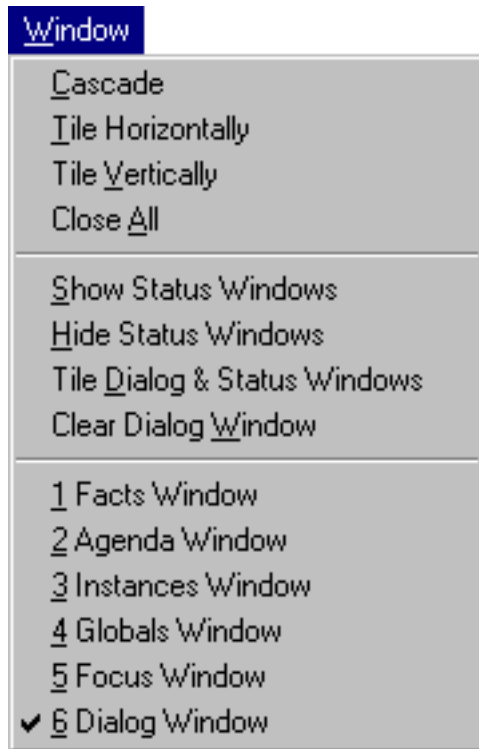
2.6.1 Cascade

Arranges windows as overlapped tiles.

2.6.2 Tile Horizontally

Arranges windows as non-overlapping tiles from top to bottom.

2.6.3 Tile Vertically

Arranges windows as non-overlapping tiles from left to right.

2.6.4 Close All

Closes all status and edit windows.

2.6.5 Show Status Windows

This command displays the **Facts Window**, **Agenda Window**, **Instances Window**, **Globals Window**, and **Focus Window** (creating them if they do not already exist). This command is available when CLIPS is executing.

2.6.6 Hide Status Windows

This command conceals the **Facts Window**, **Agenda Window**, **Instances Window**, **Globals Window**, and **Focus Window**. This command is available when CLIPS is executing.

2.6.7 Tile Dialog & Status Windows

Arranges the dialog and status windows as non-overlapping tiles.

2.6.8 Clear Dialog Window

This command clears all of the text in the **Dialog Window**. From the CLIPS command prompt, the command **clear-window** (which takes no arguments) will also clear all of the text in the **Dialog Window**. This command is available when CLIPS is executing.

2.6.9 Facts Window

This command displays the **Facts Window** if it is not already visible, otherwise it brings the window to the front. The **Facts Window** displays the list and number of facts in the fact-list. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.10 Agenda Window

This command displays the **Agenda Window** if it is not already visible, otherwise it brings the window to the front. The **Agenda Window** displays the list of activation's currently on the agenda. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.11 Instances Window

This command displays the **Instances Window** if it is not already visible, otherwise it brings the window to the front. The **Instances Window** displays the list of existing instances and their slot values. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.12 Globals Window

This command displays the **Globals Window** if it is not already visible, otherwise it brings the window to the front. The **Globals Window** displays the list of global variables and their current values. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.13 Focus Window

This command displays the **Focus Window** if it is not already visible, otherwise it brings the window to the front. The **Focus Window** displays the current focus stack. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.14 Dialog Window

This command brings the **Display Window** to the front. A check mark is placed by the window name if it is the front-most window. This command is available when CLIPS is executing.

2.6.15 Edit Windows

The list of open editing windows is displayed after the list of status and dialog windows. Selecting an editing window brings the window to the front. A check mark is placed by the editing window name if it is the front-most window.

2.7 THE HELP MENU

Help
About CLIPS...
CLIPS Help...
Command Complete... Ctrl+J
✔ Status Line
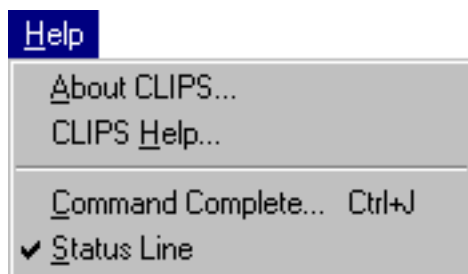
2.7.1 About CLIPS

This command displays version information about CLIPS. Click the OK button to continue. This command is available when CLIPS is executing.

2.7.2 CLIPS Help

This command displays the CLIPS hypertext help information. This command is equivalent to the CLIPS command (winhelp <topic-name>) or (winhelp). This command is available when CLIPS is executing. For more information see the Microsoft Windows User's Guide on using help.

2.7.3 Complete... (Ctrl+J)

This command "completes" the symbol currently being entered in the display window. In the display window, the symbol to be completed is always the last symbol in the display window (the current selection has no effect). If no possible completion's exist, a beep is sounded. If only one possible completion exists (e.g. "deftemplat" for "deftemplate"), then the symbol is automatically completed. If more than one completion exists, a dialog is displayed showing all possible completion's (or optionally a beep is sounded—See section 2.3.6).

The following dialog shows the possible completion's for the symbol "get-":
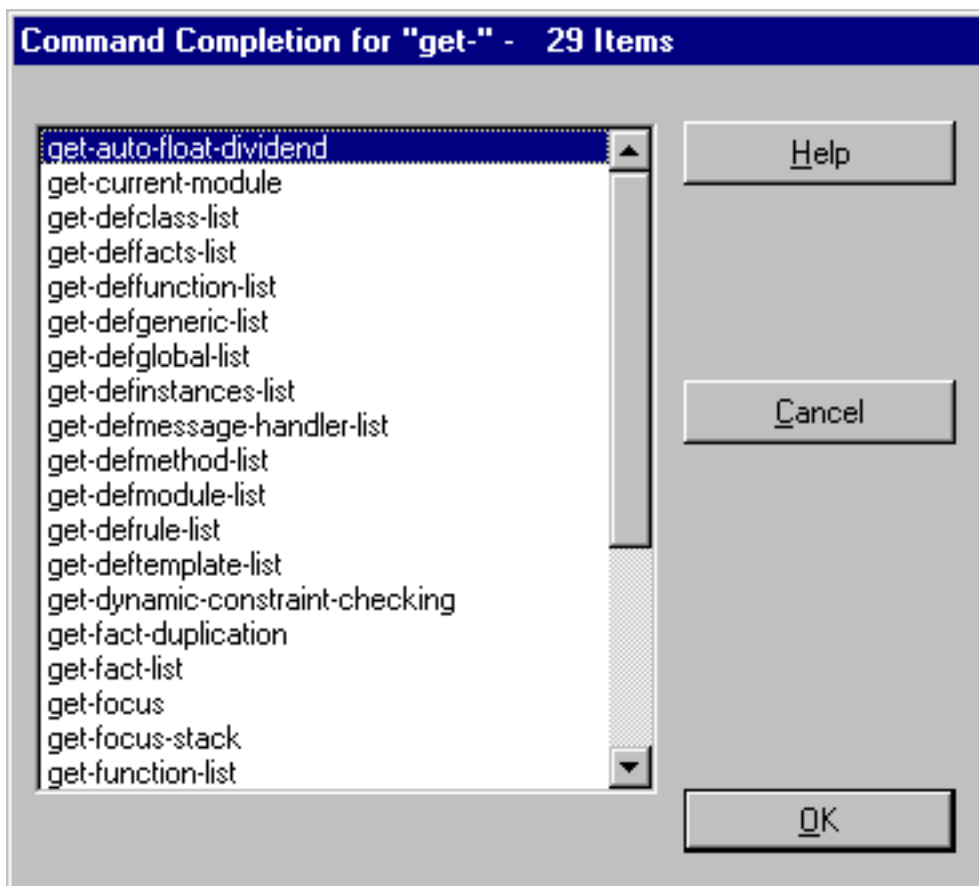
Click the **OK** button to complete the symbol with the current selection. Click the **Cancel** button to terminate the dialog without any command completion. Note that symbols defined by the user

(as part of a construct or a CLIPS data structure such as a fact or instance) will also be listed for command completion. This command is not available when CLIPS is executing.

Click the **Help** button to obtain hypertext information about the current selection. Note that some of the selections do not have an associated help entry, in this event, the help system will return a topic not found error. For more information about help see Section 2.6.2.

2.8 BUILDING THE WINDOWS XP EXECUTABLES

In order to create the Windows executables, you must install the source code using the **CLIPS 6.30 Windows Source Code Installer**. The source code (compressed in the file Projects.zip) will be installed in the directory selected for installation (typically C:\Program Files\CLIPS). Once installed, you must then extract the contents of the Projects.zip file by right clicking on it and selecting the "Extract All…" menu item.

2.8.1 Building CLIPSWin Using Microsoft Visual C++ Express 2008

From the root CLIPS directory (typically **C:\Program Files\CLIPS**) navigate to the **Projects\MVC2008** directory. Open the file **CLIPS.sln** by double clicking on it or right click on it and select the **Open** menu item. After the file opens in the Visual C++ application, right click on the **CLIPSWin** project name in the **Solution Explorer** pane and select the **Build** menu item. If the **Solution Explorer** pane is not initially visible, first select the **Solution Explorer** menu item from the **View** Menu. When compilation is complete, the executable **CLIPSWin.exe** will be in the **Projects\MVC2008** directory.

2.8.2 Building CLIPSDOS Using Microsoft Visual C++ Express 2008

From the root CLIPS directory (typically **C:\Program Files\CLIPS**) navigate to the **Projects\MVC2008** directory. Open the file **CLIPS.sln** by double clicking on it or right click on it and select the **Open** menu item. After the file opens in the Visual C++ application, right click on the **CLIPSDOS** project name in the **Solution Explorer** pane and select the **Build** menu item. If the **Solution Explorer** pane is not initially visible, first select the **Solution Explorer** menu item from the **View** Menu. When compilation is complete, the executable **CLIPSDOS.exe** will be in the **Projects\MVC2008** directory.

2.8.3 Building CLIPSWin Using Borland Turbo C++ 2006

From the root CLIPS directory (typically **C:\Program Files\CLIPS**) navigate to the **Projects\BTC2006** directory. Open the file **CLIPS.bdsgroup** by double clicking on it or right click on it and select the **Open** menu item. After the file opens in the Turbo C++ application, right click on the **CLIPSWin.exe** project name in the **Project Manager** pane and select the

Make menu item. If the **Project Manager** pane is not initially visible, first select the **Project Manager** menu item from the **View** Menu. When compilation is complete, the executable **CLIPSWin.exe** will be in the **Projects\BTC2006** directory.

2.8.4 Building CLIPSDOS Using Borland Turbo C++ 2006

From the root CLIPS directory (typically **C:\Program Files\CLIPS**) navigate to the **Projects\BTC2006** directory. Open the file **CLIPS.bdsgroup** by double clicking on it or right click on it and select the **Open** menu item. After the file opens in the Turbo C++ application, right click on the **CLIPSDOS.exe** project name in the **Project Manager** pane and select the **Make** menu item. If the **Project Manager** pane is not initially visible, first select the **Project Manager** menu item from the **View** Menu. When compilation is complete, the executable **CLIPSDOS.exe** will be in the **Projects\BTC2006** directory.

Section 3 - CLIPS Macintosh Interface

This section provides a brief summary of each menu command found in the CLIPS 6.3 Macintosh interface. The menus are listed in the left to right order in which they appear in the menu bar. The commands within each menu are also listed in the order in which they appear in the menu.

The GUI interface supports the use of multiple environments. In CLIPS, an environment is an "instance" of the CLIPS engine that can load and execute CLIPS programs. Typically you will only use one environment when developing a program—the one that is automatically created when the CLIPS IDE application launches—but situations can occur where having multiple environments at the same time is useful; For example, a side by side comparison of two different versions of the same program.

In addition to the browsers for examining the current CLIPS state, the environment windows provide a command prompt that allows commands to be entered in a manner similar to the standard CLIPS command line interface. Any CLIPS I/O which is directed to standard input or standard output is also handled through the environment window. The initial environment window displayed when the CLIPS IDE is launched looks like the following:
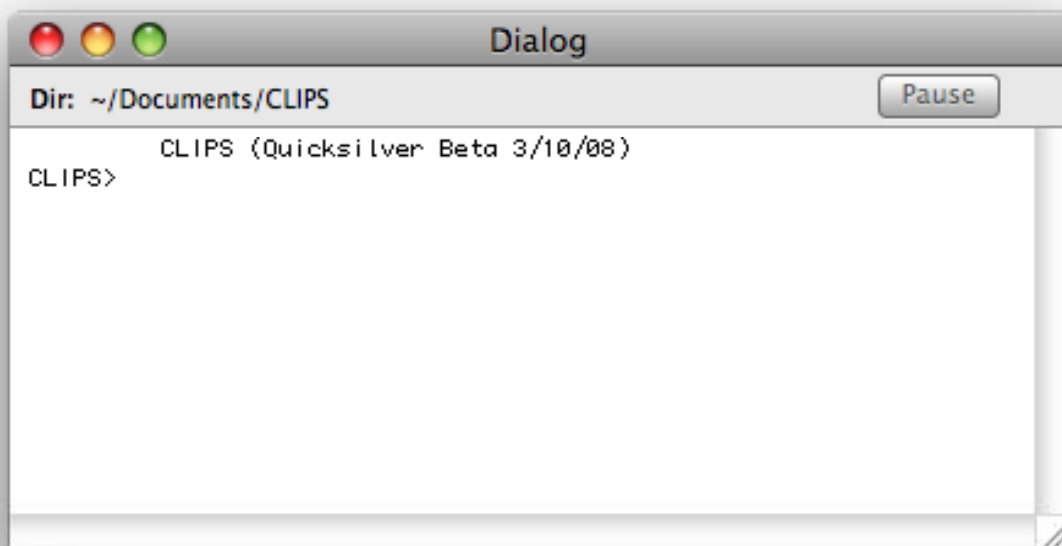
The title of all environment windows begins with Dialog. A status bar is displayed beneath the title bar. On the left side of the status bar is the current working directory. A **Pause** button is on the right side of the status bar. The CLIPS IDE is multi-threaded with each environment generating a separate thread to execute commands. Pressing the **Pause** button while a command is executing will suspend execution of the command thread. This is useful if you need to examine

the output of the executing program. Pressing the **Pause** button a second time will resume execution of the command thread.

The interface also provides a text editor for writing CLIPS programs. Editing windows look like the following:
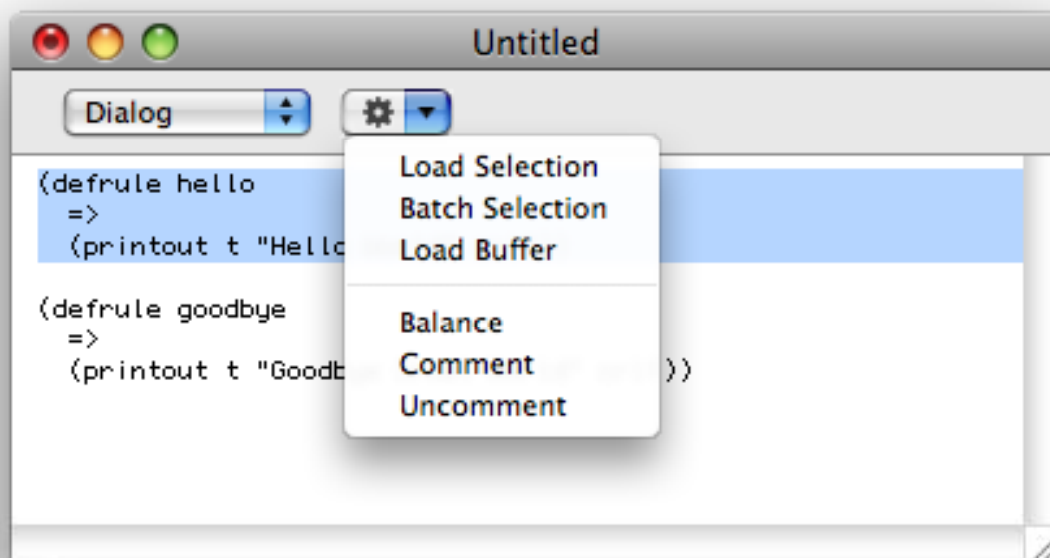
Newly created editing windows begin with the word Untitled in their title bar. If an editing window is associated with a file, then the title bar will contain the file name. Beneath the title bar is a control strip. The drop-down menu on the left side of the strip allows the environment window to which the editing window is "attached" to be changed. The action menu to the right of the attachment menu provides access to the same menu items that are in the **Text** menu. In the window shown previously, selecting the **Load Selection** menu item (either from the action menu or the **Text** menu) would load the selection in the editing window in the *Dialog* environment window.

3.1 THE CLIPS IDE MENU
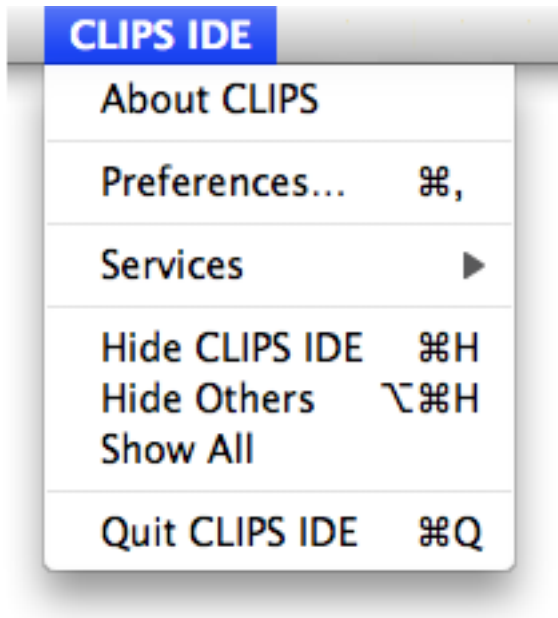
3.1.1 About CLIPS

This command displays version information about the CLIPS IDE application.

3.1.2 Preferences... (⌘,)

This command displays a dialog box which allows the user to set the parameters for several options in the CLIPS Macintosh interface.

3.1.2.1 Watch Tab

The Watch tab allows you to specify the initial settings for CLIPS watch items when a new environment window is created.
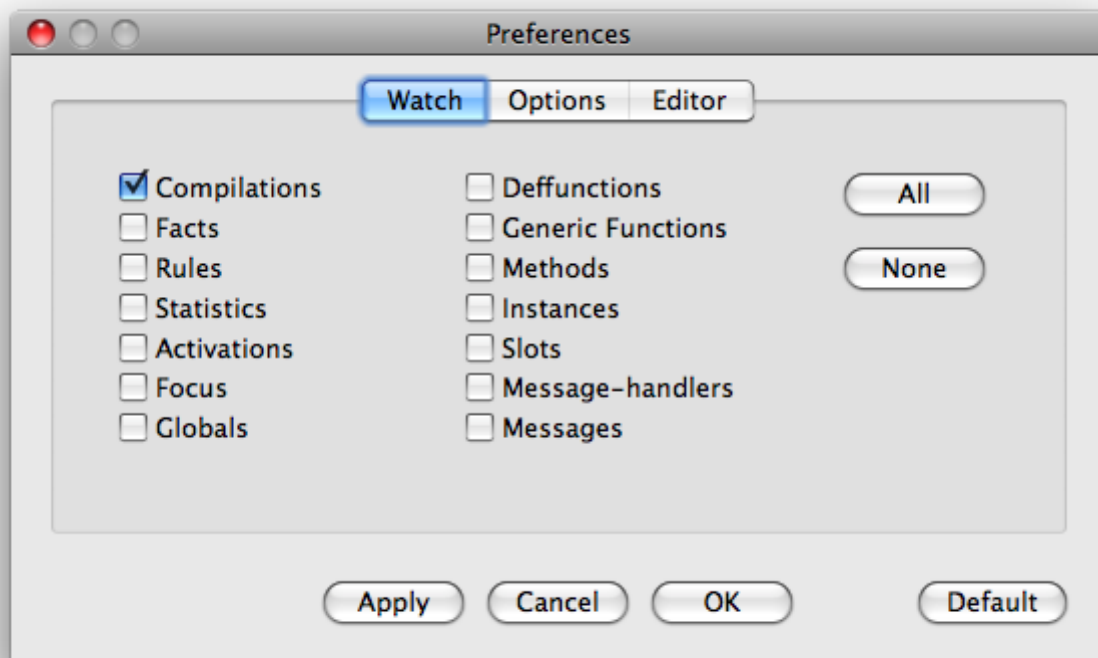
Watch items can be enabled or disabled by clicking the appropriate check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item check boxes. Pressing the **None** button unchecks all of the watch item check boxes. Clicking the **OK** or **Apply** button exits the dialog and uses the current watch settings for new environment windows. Clicking the **Apply** button also saves the settings to be used the next time the CLIPS IDE is launched. Clicking the **Cancel** button exits the dialog, but retains the original settings of the watch items before the dialog was entered. Clicking the **Default** button restores the default settings for the watch items.

3.1.2.2 Options Tab

The Options tab allows the user to set initial settings for various CLIPS execution options when a new environment window is created.

The **Salience Evaluation** pop-up menu allows the current salience evaluation behavior to be changed (to either when-defined, when-activated, or every-cycle). The **Strategy** pop-up menu allows the current conflict resolution strategy to be changed (to either depth, breadth, lex, mea, complexity, simplicity, or random). The **Static Constraint Checking**, **Dynamic Constraint Checking**, **Reset Global Variables**, **Sequence Expansion Operator Recognition**, **Incremental Reset**, **Auto-Float Dividend**, and **Fact Duplication** check boxes are used to enable and disable the named CLIPS option. Enabled options have a check in their check box while disabled items have none. The **Incremental Reset** option cannot be changed if any defrules are currently defined. Clicking the **OK** or **Apply** button exits the dialog and uses the current execution option settings for new environment windows. Clicking the **Apply** button saves the settings to be used the next time the CLIPS IDE is launched. Clicking the **Cancel** button exits the dialog, but retains the original settings of the CLIPS execution options before the dialog was entered. Clicking the **Default** button restores the default settings for the execution options.

3.1.2.3 Editor Tab

The Editor tab allows text options for the environment and editing windows to be set.
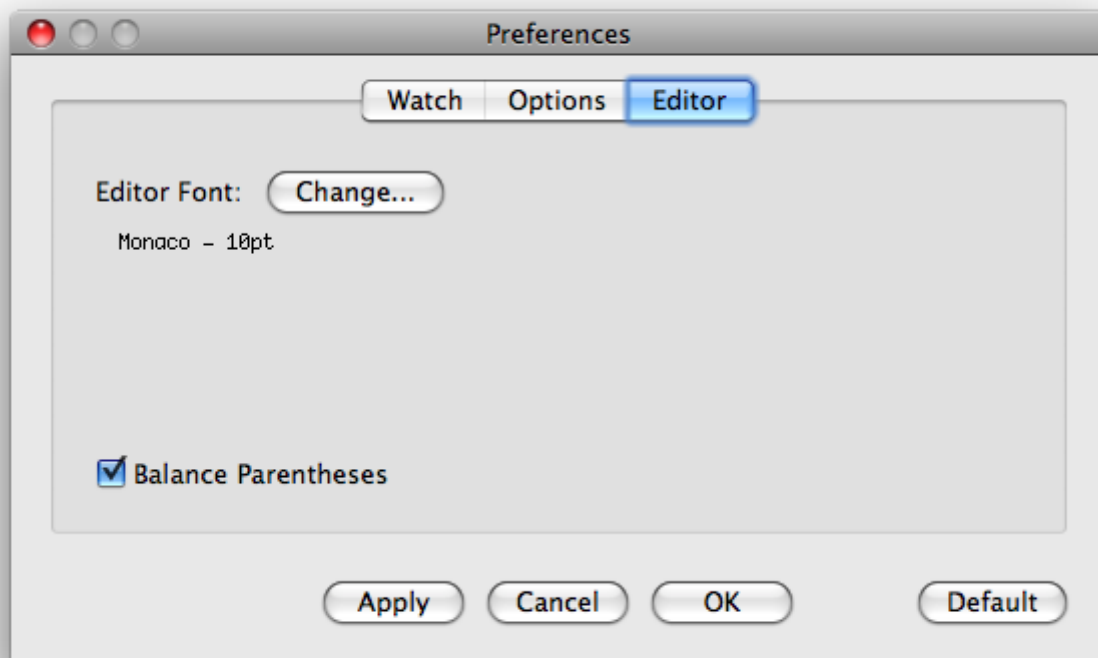
The **Change…** button allows the font used in the environment and editing windows to be changed. When you click this button, a Fonts dialog will appear. Select a font or font size from the Font dialog and the text in the Editor tab will change to reflect the new font or font size.

The **Balance Parentheses** check box, if enabled, causes the matching left parenthesis to be momentarily highlighted whenever a right parenthesis is type or the cursor is moved immediately after a right parenthesis.

Clicking the **OK** or **Apply** button exits the dialog and changes the editor settings to those shown in the dialog. Clicking the **Apply** button also saves the settings to be used the next time the CLIPS IDE is launched. Clicking the **Cancel** button exits the dialog, but retains the original editor settings before the dialog was entered. Clicking the **Default** button restores the default editor settings.

3.1.3 Services

This command provides access to system services.

3.1.4 Hide CLIPS IDE (⌘H)

This commands hides all of the windows associated with the CLIPS IDE application.

3.1.5 Hide Others (⌥⌘H)

This command hides all of the windows associated with other applications.

3.1.6 Show All

This command shows all of the windows associated with other applications.

3.1.7 Quit CLIPS IDE (⌘Q)

This command causes the CLIPS IDE to quit. The user will be prompted to save any files with unsaved changes.

3.2 THE FILE MENU
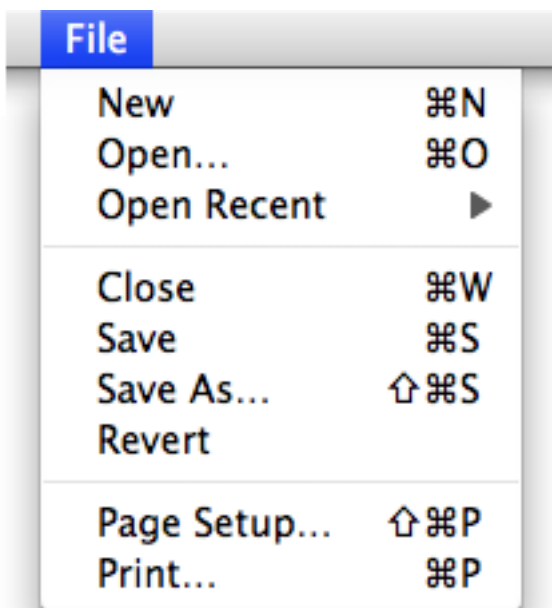
3.2.1 New (⌘N)

This command opens a new buffer for editing with the window name Untitled.

3.2.2 Open... (⌘O)

This command displays the standard file selection dialog sheet, allowing the user to select a text file to be opened as a buffer for editing. More than one file can be opened at the same time, however, the same file cannot be opened more than once. As files are opened, they are automatically stacked.

3.2.3 Open Recent

This command displays a list of recently opened files, allowing the user to select a text file to be opened as a buffer for editing.

3.2.4 Close (⌘W)

This command closes the active window if it has red close button in its upper left corner. If the active window is an editing buffer that has been modified since it was last saved, an alert sheet will confirm whether the changes should be saved or discarded or whether the close command should be canceled.

3.2.5 Save (⌘S)

This command saves the file in the active edit window. If the file is untitled, a save file dialog sheet will prompt for a file name under which to save the file.

3.2.6 Save As... (⇧⌘S)

This command allows the active edit window to be saved under a new name. A save file dialog sheet will appear to prompt for the new file name. The name of the editing window will be changed to the new file name.

3.2.7 Revert

This command restores the active edit window to the last-saved version of the file in the buffer. Any changes made since the file was last saved will be discarded.

3.2.8 Page Setup... (⇧⌘P)

This command allows the user to specify information about the size of paper used by the printer.

3.2.9 Print... (⌘P)

This command allows the user to print the active edit window.

3.3 THE EDIT MENU
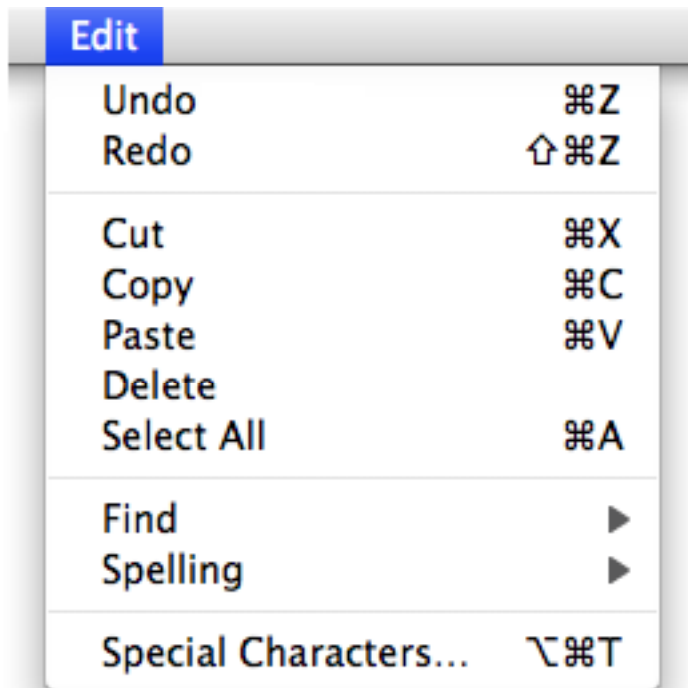
3.3.1 Undo (⌘Z)

This command allows you to undo your last editing operation. Typing, cut, copy, paste, and delete operations can all be undone. The **Undo** menu item will change in the **Edit** menu to reflect the last operation performed. For example, if a **Paste** command was just performed, the **Undo** menu item will read **Undo Paste**.

3.3.2 Redo (⇧⌘Z)

This command allows you to redo your last editing operation. Typing, cut, copy, paste, and delete operations can all be redone. The **Redo** menu item will change in the **Edit** menu to reflect the last operation performed. For example, if a **Paste** command was just performed, the **Redo** menu item will read **Undo Paste**.

3.3.3 Cut (⌘X)

This command removes selected text in the active edit window and places it in the Clipboard.

3.3.4 Copy (⌘C)

This command copies selected text in the active edit window or the display window and places it in the Clipboard.

3.3.5 Paste (⌘V)

This command copies the contents of the Clipboard to the selection point in the active edit window or the display window. If selected text is in the active edit window, it is replaced by the contents of the Clipboard. Pasted text to the display window is always placed at the end of the display window regardless of whether there is selected text.

3.3.6 Delete

This command removes selected text in the active edit window. The selected text is not placed in the Clipboard.

3.3.7 Select All (⌘A)

This command selects all of the text in the active edit or display window.

3.3.8 Find
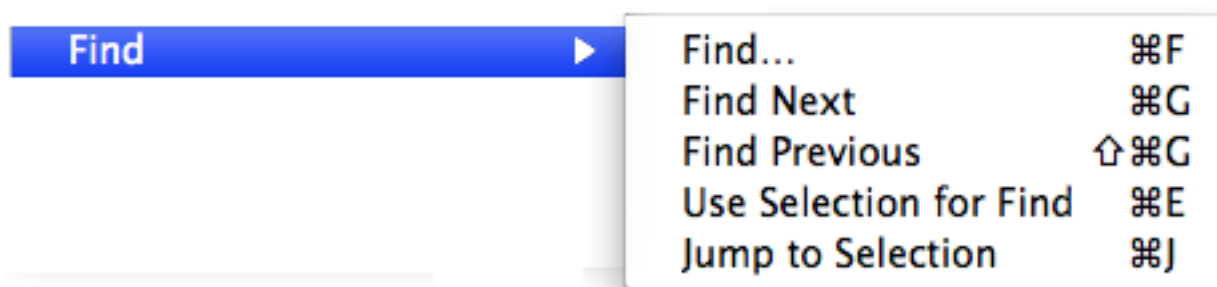
3.3.8.1 Find... (⌘F)

This command displays a dialog box which allows the user to set parameters for text search and replacement operations. The dialog box that appears allows a search and replacement string to be specified.
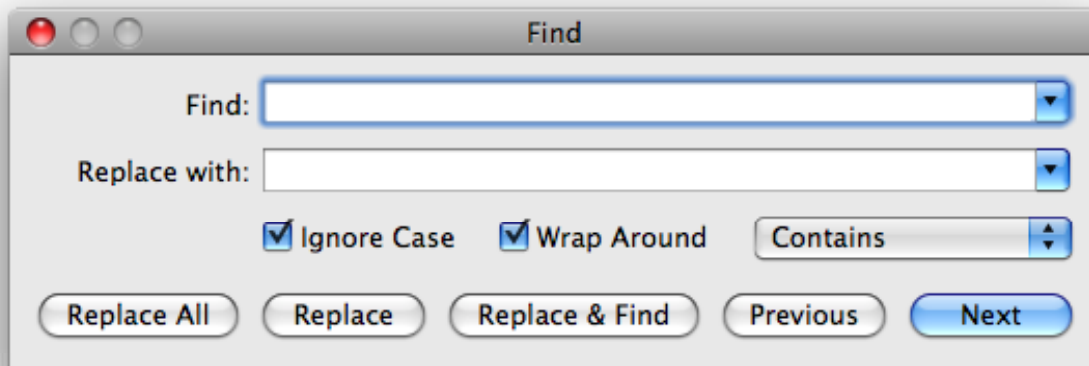
Three other options can be set in the search dialog box. The **Ignore Case** option makes the string search operation case-insensitive for alphabetic characters; that is, the string "Upper" will match the string "uPPER". The **Wrap Around** option determines whether the search is restarted at the top of the document when the bottom of the document is reached. The drop-down menu allows the match criterion to be set. If it is set to **Contains**, then the search matches any text containing the search string. If it is set to **Starts with**, then only whole words beginning with the search string will be matched. If it is set to **Full word**, then only whole words will be matched.

Once search options have been set, one of five search dialog buttons can be pressed. The **Replace All** button replaces all occurrences of the **Find** string with the **Replace With** string. The **Replace** button replaces the current selection with the **Replace With** string. The **Replace & Find** button replaces the current selection with the **Replace With** string and finds and selects the next match for the **Find** string. The **Previous** button finds and selects the previous match for the **Find** string. The **Next** button finds and selects the next match for the **Find** string.

3.3.8.2 Find Next (⌘G)

This command finds and selects the next match for the **Find** string.

3.3.8.3 Find Previous (⇧⌘G)

This command finds and selects the previous match for the **Find** string.

3.3.8.4 Use Selection for Find (⌘E)

This command sets the **Find** string to the current selection.

3.3.8.5 Jump to Selection (⌘J)

This command brings the current selection into view.

3.3.9 Spelling
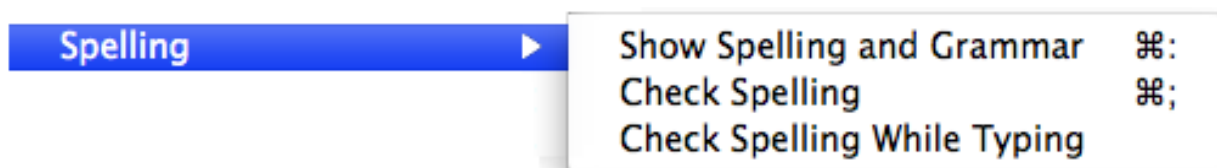
3.3.9.1 Show Spelling and Grammar (⌘:)

This command displays the Spelling and Grammar modeless dialog box which allows you to check the spelling and grammar of the front most editing window.

3.3.9.2 Check Spelling (⌘;)

This command finds and selects the next misspelled word in the front most editing window. The misspelled word is underlined in red. Control-clicking on a misspelled word will display a menu containing a list of suggestions to correct the misspelling.

3.3.9.3 Check Spelling While Typing

This command enables/disables automatic spell checking. If enabled, spelling is automatically check as words are typed with misspelled words underlined in red. Control-clicking on a misspelled word will display a menu containing a list of suggestions to correct the misspelling.

3.3.10 Special Characters… (⌥⌘T)

This command displays the Character modeless dialog box which allows symbols and characters from other languages to be entered as input.

3.4 THE TEXT MENU
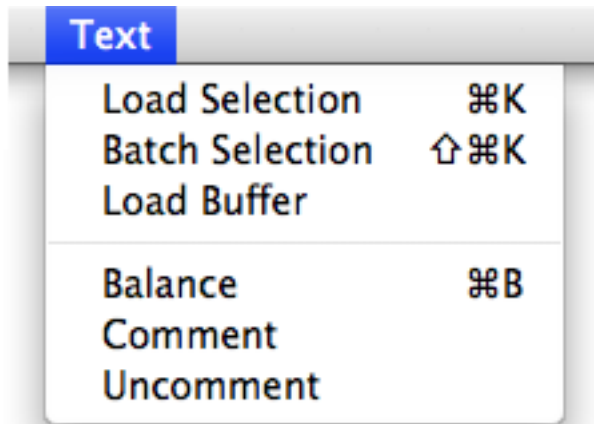
3.4.1 Load Selection (⌘K)

This command loads the current selection in the active edit window into the CLIPS knowledge base. Standard error detection and recovery routines used to load constructs from a file are also used when loading a selection (i.e., if a construct has an error in it, the rest of the construct will be skipped over until another construct to be loaded is found).

3.4.2 Batch Selection (⇧⌘K)

This command treats the current selection in the active edit window as if it were a batch file and executes it as a series of commands. Standard error detection and recovery routines used to load construct from a file are *not* used when batching a selection (i.e., if a construct has an error in it, a number of ancillary errors may be generated by subsequent parts of the same construct following the error).

3.4.3 Load Buffer

This command loads the contents of the active edit window into the CLIPS knowledge base. It is equivalent to selecting the entire buffer and executing a **Load Selection** command.

3.4.4 Balance (⌘B)

This command operates on the current selection in the active edit window by attempting to find the smallest selection containing the current selection which has balanced parentheses. Repeatedly using this command will select larger and larger selections of text until a balanced selection

cannot be found. The balance command is a purely textual operation and can be confused by parentheses found in strings.

3.4.5 Comment

This command operates on the current selection in the active edit window by adding a semicolon to the beginning of each line contained in the selection.

3.4.6 Uncomment

This command operates on the current selection in the active edit window by removing a semicolon (if one exists) from the beginning of each line contained in the selection.

3.5 THE ENVIRONMENT MENU
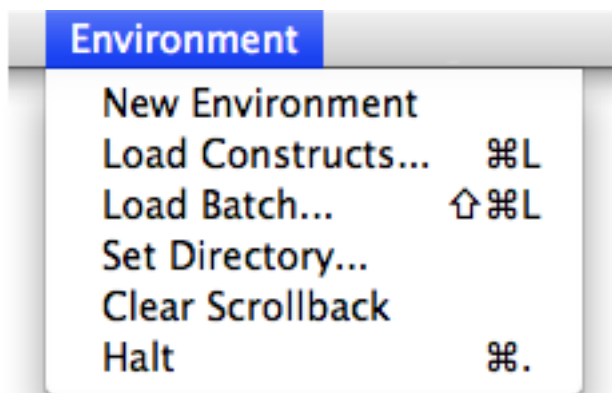
3.5.1 New Environment

This command creates a new environment window. Environments are separate from each other, can have different programs loaded, and can be executed concurrently.

3.5.2 Load Constructs... (⌘L)

This command displays the standard file selection dialog sheet, allowing the user to select a text file to be loaded into the knowledge base. This command is equivalent to the CLIPS command (load <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS load command will be echoed to the environment window and executed.

3.5.3 Load Batch...

This command displays the standard file selection dialog sheet, allowing the user to select a text file to be executed as a batch file. This command is equivalent to the CLIPS command (batch <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS batch command will be echoed to the environment window and executed.

3.5.4 Set Directory…

This command displays the standard folder selection dialog sheet, allowing the user to select the current directory associated with the environment. CLIPS file commands such as load, batch, and open use the current directory to determine the location where file operations should occur. The current directory for an environment window is displayed in the status pane below the window title.

3.5.5 Clear Scrollback

This command clears all of the text in the dialog window. From the CLIPS command prompt, the command **clear-window** (which takes no arguments) will also clear all of the text in the environment window.

3.5.6 Halt (⌘.)

If rules are executing, this command halts execution when the currently executing rule has finished. Holding down the option key changes this menu item to Halt Immediately which will halt rule execution after the current RHS action.

3.6 THE DEBUG MENU
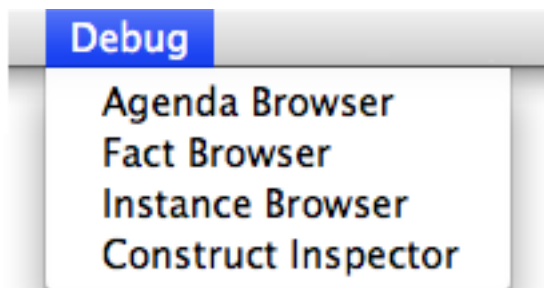
3.6.1 Agenda Browser

The **Agenda Browser** allows the activations on the agenda to be examined. The list on the left side of the window shows the modules currently on the focus stack. The list of the right side of the window shows the activations on the agenda of the selected module from the focus stack. The drop-down menu in the upper left of the browser allows the environment window to which the browser is "attached" to be changed.
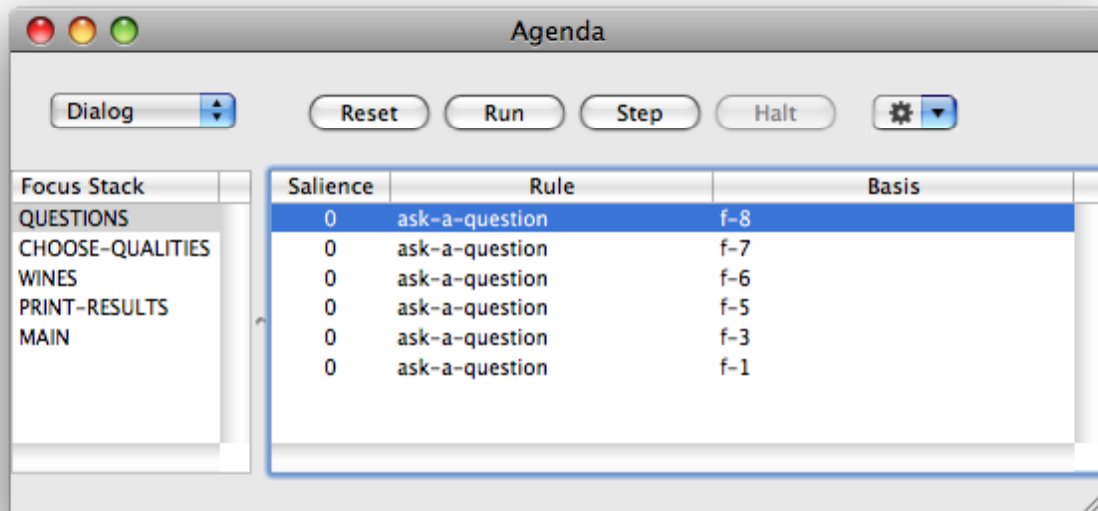
The **Reset** button sends a "(reset)" command to the attached environment window. The **Run** button sends a "(run)" command to the attached environment window. The **Step** button sends a "(run 1)" command to the attached environment window. Pressing the **Halt** button when rules are executing will halt execution when the currently executing rule has finished.

Functionality for the items in the cog drop down menu in the upper right has not been implemented.

3.6.2 Fact Browser

The **Fact Browser** allows the facts in the fact list to be examined. The list on the left side of the window shows the modules currently defined. The list in the middle of the window shows the facts that are visible to the selected module from the module list. The list on the right side of the window shows the slot values of the selected fact from the fact list. The drop-down menu in the upper left of the browser allows the environment window to which the browser is "attached" to be changed.
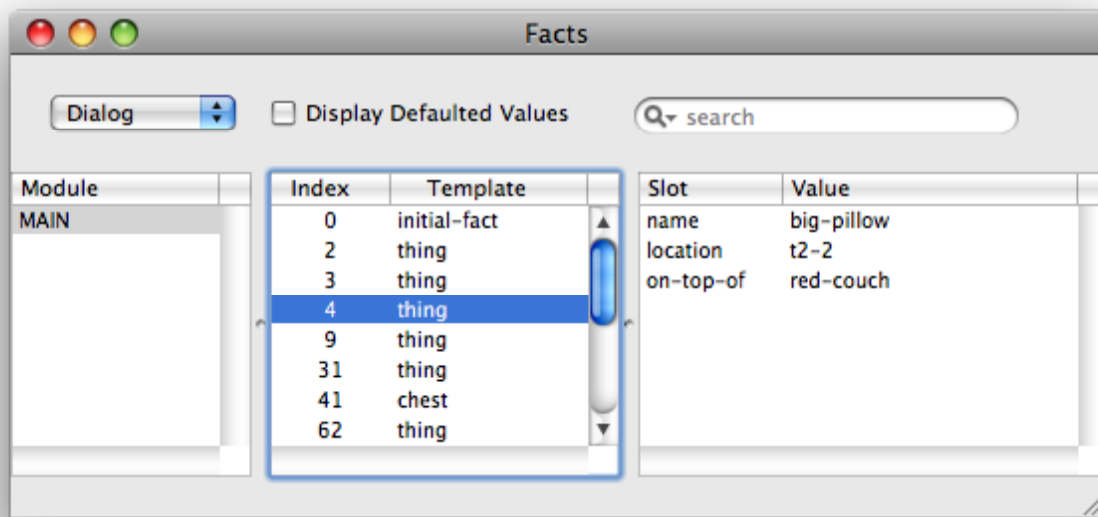
The list of facts can be sorted based on either the fact index or the associated deftemplate name by clicking on either the **Index** or the **Template** column header. The list of slots can be sorted based on either the slot name or the slot value by clicking on either the **Slot** or the **Value** column header. If the **Display Defaulted Values** checkbox is enabled, then all of the slots of the selected fact will be displayed. If the checkbox is disabled, then only those slots that have a value different from their default slot value will be displayed.

The search text field can be used to filter the facts that are displayed in the fact list. When search text is entered and the return key is pressed each fact and its slots are examined to determine if the search text is found within one of the following templates:

```
f-<index>
<deftemplate-name> <slot-name> <slot-value>
```

For example, if the fact associated with the deftemplate thing had a fact index of 4 and slots name with value big-pillow, location with value t2-2, and on-top-of with value red-couch, then the fact would be displayed in the fact list only if the search text was found in one of the following strings:

```
f-4
thing name big-pillow
thing location t2-2
thing on-top-of red-couch
```

3.6.3 Instance Browser

The **Instance Browser** allows the instances in the instance list to be examined. The list on the left side of the window shows the modules currently defined. The list in the middle of the window shows the instances that are visible to the selected module from the module list. The list on the right side of the window shows the slot values of the selected instance from the instance list. The drop-down menu in the upper left of the browser allows the environment window to which the browser is "attached" to be changed.
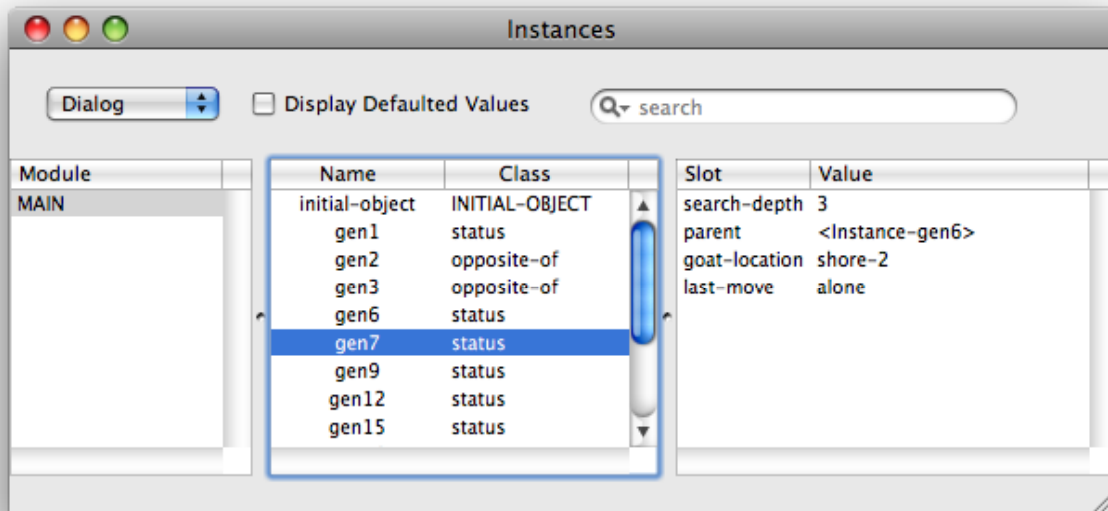
The list of instances can be sorted based on either the instance name or the associated defclass name by clicking on either the **Name** or the **Class** column header. The list of slots can be sorted based on either the slot name or the slot value by clicking on either the **Slot** or the **Value** column header. If the **Display Defaulted Values** checkbox is enabled, then all of the slots of the selected instance will be displayed. If the checkbox is disabled, then only those slots that have a value different from their default slot value will be displayed.

The search text field can be used to filter the instances that are displayed in the instance list. When search text is entered and the return key is pressed each instance and its slots are examined to determine if the search text is found within one of the following templates:

```
[<name>]
<defclass-name> <slot-name> <slot-value>
```

For example, if the instance associated with the defclass THING had the instance name [thing1] and slots name with value big-pillow, location with value t2-2, and on-top-of with value red-couch, then the instance would be displayed in the instance list only if the search text was found in one of the following strings:

```
[thing1]
THING name big-pillow
THING location t2-2
THING on-top-of red-couch
```

3.6.4 Construct Inspector

The **Construct Inspector** floats above the other CLIPS IDE windows and changes to show the text of the associated construct when one of the items from the browsers is selected.

```
(deftemplate MAIN::thing
    (slot name (type SYMBOL) (default ?NONE))
    (slot location (type SYMBOL) (default ?NONE))
    (slot on-top-of (type SYMBOL) (default floor))
    (slot weight (type SYMBOL) (allowed-symbols light heavy) (default light)))
```
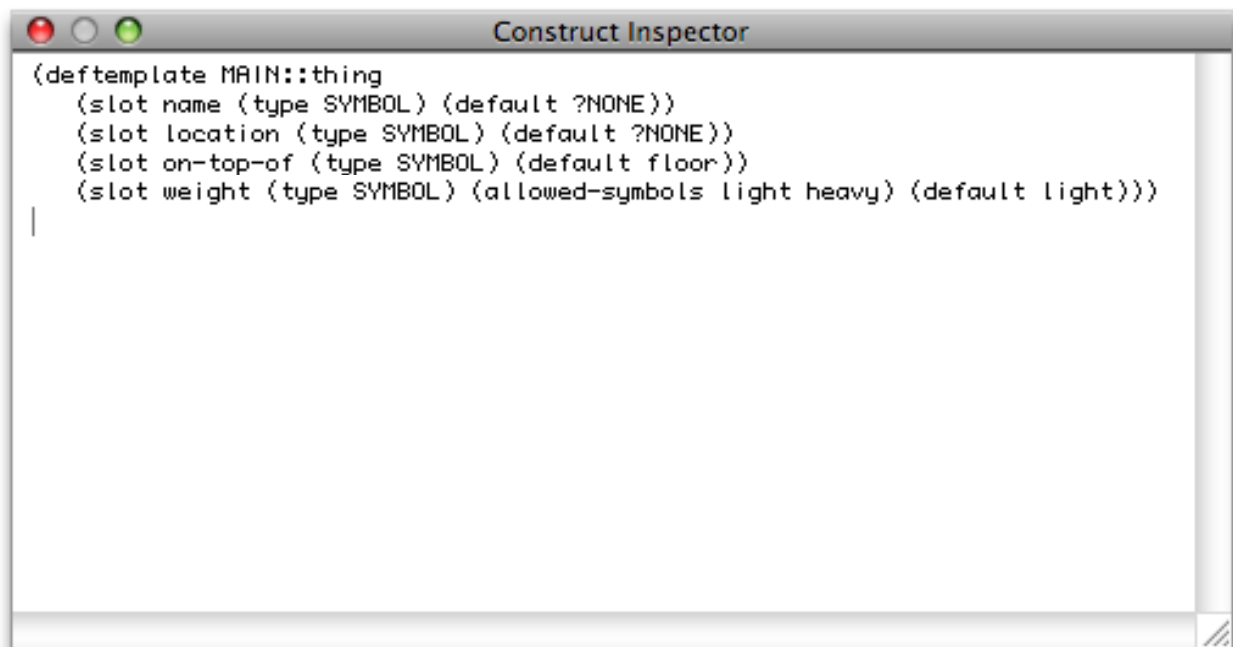
3.7 THE WINDOW MENU
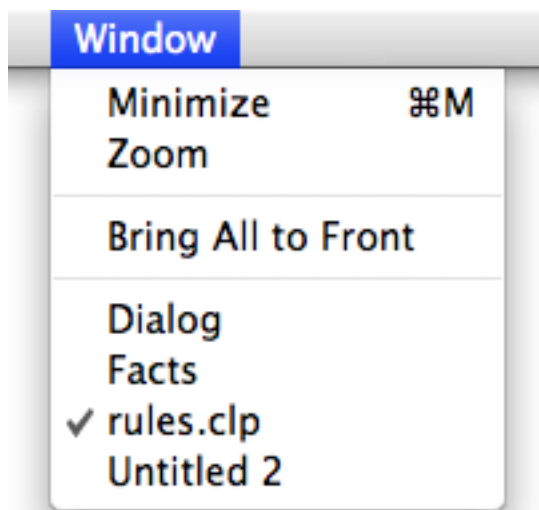
3.7.1 Minimize (⌘M)

Minimizes the front-most window placing it in the dock.

3.7.2 Zoom

Enlarges the front-most window or, if the window has already been enlarged, shrinks the window back to its original size.

3.7.3 Bring All to Front

Brings all of the windows associated with the CLIPS IDE to the front most layer of application windows. That is, all of the CLIPS IDE windows will be above the windows of all other applications.

3.7.4 Window List

The bottom portion of the Window menu (everything below the All Above menu item) is a list of all windows associated with the CLIPS IDE. A check mark is placed by the window name to indicate that it is the frontmost window. A filled circle appears next to an edit window title that has changes that need to be saved (unless it is the frontmost window).

3.8 THE HELP MENU
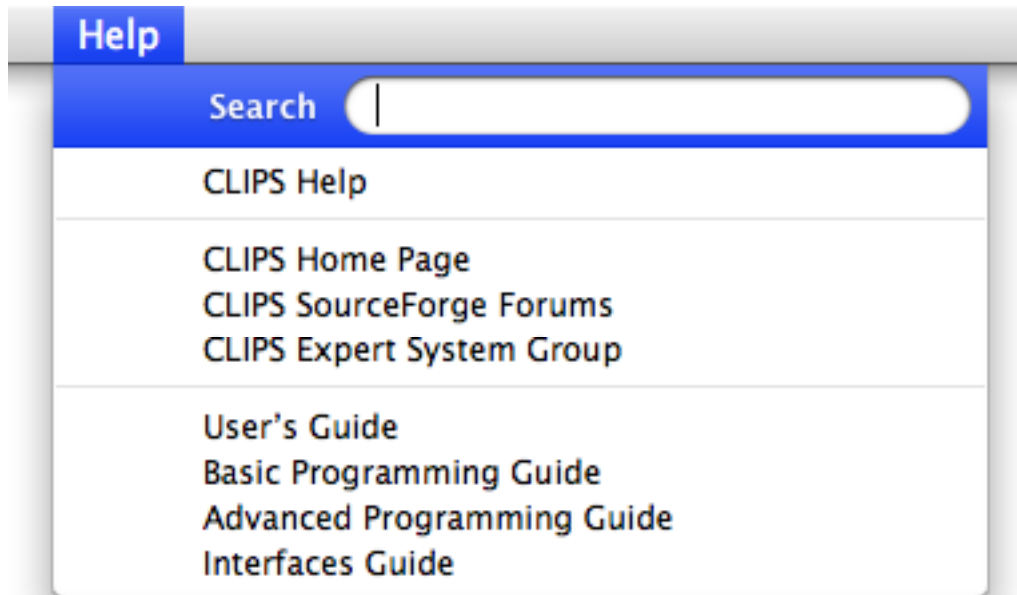
3.8.1 CLIPS Help

Not currently supported.

3.8.2 CLIPS Home Page

Opens the CLIPS Home web page on SourceForge.

3.8.3 CLIPS SourceForge Forums

Opens the CLIPS Discussion Forums web page on SourceForge.

3.8.4 CLIPS Expert System Group

Opens the CLIPS Expert System Group web page on Google Groups.

3.8.5 User's Guide

Opens a web page containing the CLIPS User's Guide.

3.8.6 Basic Programming Guide

Opens a web page containing the CLIPS Basic Programming Guide.

3.8.7 Advanced Programming Guide

Opens a web page containing the CLIPS Advanced Programming Guide.

3.8.8 Interfaces Guide

Opens a web page containing the CLIPS Interfaces Guide.

3.9 CREATING THE MACINTOSH EXECUTABLES

In order to create the Mac OS X executables, you must install the source code using the **CLIPS.pkg** Package Installer. The source code (compressed in the file **Source Code.zip**) will be installed in the directory selected for installation (typically :Applications:CLIPS). Once installed, you must then extract the contents of the **Source Code.zip** file by double clicking on it.

3.9.1 Building CLIPS Using Xcode 3.0

From the root CLIPS directory (typically **:Applications:CLIPS**) navigate to the **Source Code** directory. Open the file **CLIPS.xcodeproj** by double clicking on it. After the file opens in the Xcode application, select the **Build** menu item from the **Build** menu to create the CLIPS IDE executable. Alternately, from the **Project** menu select the **Set Active Target** menu item and choose CLIPS Console, then select the **Build** menu item from the **Build** menu to create the CLIPS Console executable. The generate executables can be found in the **:build:Release** directory.

Section 4 - CLIPS X Window Interface

This section provides a brief summary of the CLIPS 6.3 X Window interface (**xclips**) for systems running X Windows. The menus are listed in the left to right order in which they appear in the menu bar. The commands within each menu are also listed in the order in which they appear in the menu.

The rest of the **xclips** is straightforward. Commands can be entered directly in the dialog window in a fashion similar to the standard CLIPS command line interface.

By default, **xclips** is black and white. A resource file, **Xclips**, is provided to add color to the interface. By default X will look for this resource file in the user's home directory. A user can modify his or her resource file if he or she wishes; however, it could be more convenient if he or she uses the provided color utility, **color**. The color utility has to be in the same directory with **xclips**, so it could be executed by **xclips** (see section 4.5.8).

The execution of the current command or CLIPS program can be halted by pressing Ctrl-H

Some menu commands are available while CLIPS is executing a user program. Among these commands are the **Options...** menu item, the **Watch...** menu item, the **Dribble...** menu item, and all menu items in **Window** menu. The command summary for each menu item indicates whether it is available while CLIPS is executing. In addition, windows may be moved and resized while CLIPS is executing.

4.1 THE ![icon] MENU

This command displays version information about CLIPS. Click the left mouse button in this window to kill it. This command is not available when CLIPS is executing.

4.2 THE FILE MENU

```
┌─────────────────────────┐
│          File           │
├─────────────────────────┤
│ Edit...            ^V    │
│ Complete...        ^C    │
├─────────────────────────┤
│ Load...            ^L    │
│ Load Batch...            │
│ Load Binary...           │
│ Dribble...         ^D    │
├─────────────────────────┤
│ Save Binary...           │
├─────────────────────────┤
│ Quit              ^Q     │
└─────────────────────────┘
```

4.2.1 Edit... (^V)

This command displays a file selection box, allowing the user to select a text file to edit. A new window will be opened for each file (see section 4.6 for more details). This command is not available when CLIPS is executing.

4.2.2 Complete... (^C)

This command "completes" the symbol currently being entered in the display window. The symbol to be completed is always the last symbol in the display window. If no possible completions exist, a beep is sounded. If only one possible completion exists (e.g. "deftemplat" for "deftemplate"), then the symbol is automatically completed. If more than one completion exists, a dialog is displayed showing all possible completions. The following dialog shows the possible completions for the symbol "get":
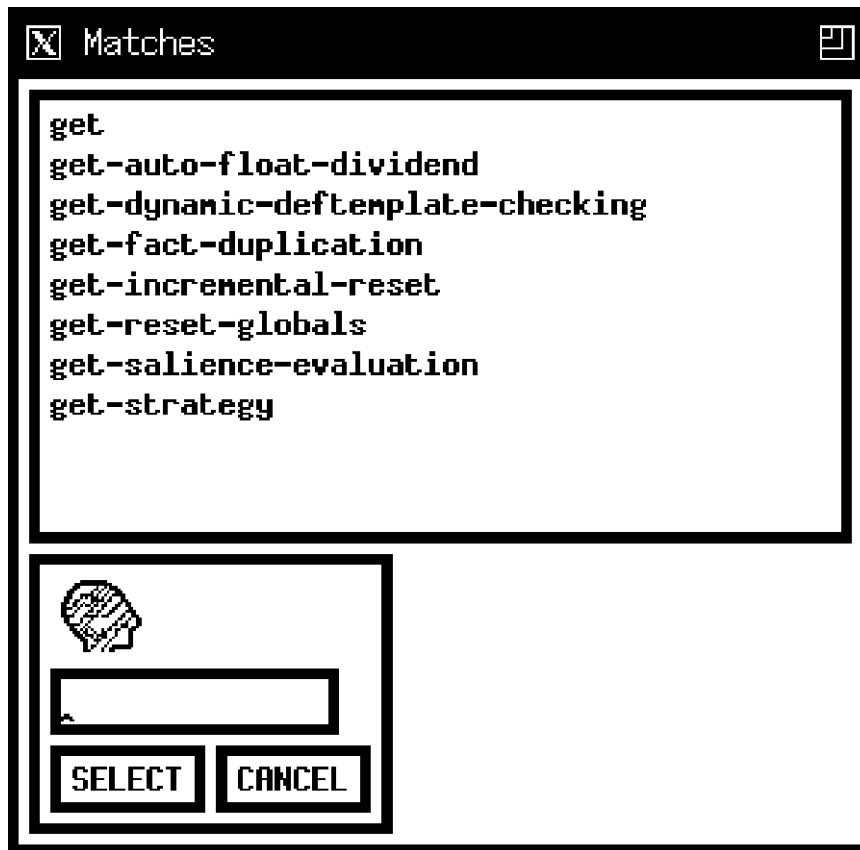
Click the **Select** button to complete the symbol with the current selection. Click the **Cancel** button to terminate the dialog without any command completion. Note that symbols defined by the user (as part of a construct or a CLIPS data structure such as a fact or instance) will also be listed for command completion. This command is not available when CLIPS is executing.

4.2.3 Load... (^L)

This command displays a file selection box, allowing the user to select a text file to be loaded into the knowledge base. This command is equivalent to the CLIPS command (load <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS load command will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.2.4 Load Batch...

This command displays a file selection box, allowing the user to select a text file to be executed as a batch file. This command is equivalent to the CLIPS command (batch <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS batch command will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.2.5 Load Binary...

This command displays a file selection box, allowing the user to select a file to be loaded as a binary image. This command is equivalent to the CLIPS command (bload <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS bload command will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.2.6 Dribble... (^N)

This command displays a file selection box, allowing the user to select a text file in which subsequent display window output will be stored. This command is equivalent to the CLIPS command (dribble-on <file-name>). When this command is chosen and a file is selected, the appropriate CLIPS dribble-on command will be echoed to the main window and executed. While the dribble file is active, this menu command will be modified to contain an X Window logo in the left margin beside the command. Selecting the menu command at this time will close the dribble text file and remove the X Window logo. This is equivalent to the CLIPS command (dribble-off). This command is available when CLIPS is executing (however the commands will not be echoed to the dialog window).

4.2.7 Save Binary...

This command allows the constructs currently stored in CLIPS to be saved as a binary image. A dialog box will appear to prompt for the new file name in which to store the binary image. This command is equivalent to the CLIPS command (bsave <file-name>). When this command is chosen and a file is entered, the appropriate CLIPS bsave command will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.2.8 Quit (^Q)

This command exits CLIPS. After choosing this command a Restart option is prompted. This option will restart CLIPS using any new color resources created by the color utility (see section 4.5.8). This command is not available when CLIPS is executing.

4.3 THE EXECUTION MENU

```
┌─────────────────────┐
│     Execution       │
├─────────────────────┤
│ Reset          ^E   │
│ Run            ^R   │
│ Step           ^T   │
│ Watch...            │
│ Options...          │
├─────────────────────┤
│ Clear CLIPS    ^K   │
│ Clear Window   ^N   │
└─────────────────────┘
```
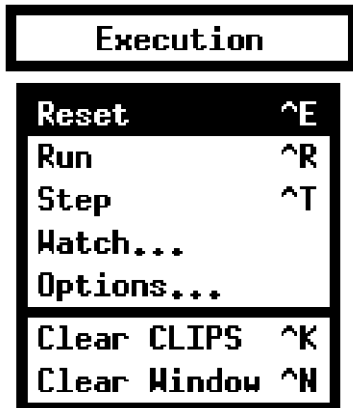
4.3.1 Reset (^E)

This command is equivalent to the CLIPS command (reset). When this command is chosen, the CLIPS command (reset) will be echoed to the main window and executed. If activations are currently on the agenda, a dialog box will appear issuing a warning and providing an opportunity to cancel this command. This command is not available when CLIPS is executing.

4.3.2 Run (^R)

This command is equivalent to the CLIPS command (run). When this command is chosen, the CLIPS command (run) will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.3.3 Step (^T)

This command is equivalent to the CLIPS command (run 1). When this command is chosen, the CLIPS command (run 1) will be echoed to the main window and executed. This command is not available when CLIPS is executing.

4.3.4 Watch...

This command displays a dialog box which allows the user to set various watch items as enabled or disabled.

```
┌─────────────────────────────────────────────────────────────┐
│  ┌──┐                Watch Menu                   ┌─┐ ┌─┐     │
│  └──┘                                             └─┘ └─┘     │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │      Compilations       │  │       Deffunctions      │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │         Facts           │  │    Generic Functions    │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │         Rules           │  │        Methods          │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │       Statistics        │  │       Instances         │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │      Activations        │  │         Slots           │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │         Focus           │  │    Messages-handlers    │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│  ┌─────────────────────────┐  ┌─────────────────────────┐    │
│  │        Globals          │  │       Messages          │    │
│  └─────────────────────────┘  └─────────────────────────┘    │
│                                                              │
│   ┌─────────────────────┐      ┌─────────────────────┐       │
│   │        All          │      │        None         │       │
│   └─────────────────────┘      └─────────────────────┘       │
│   ┌─────────────────────┐      ┌─────────────────────┐       │
│   │        Okay         │      │       Cancel        │       │
│   └─────────────────────┘      └─────────────────────┘       │
└─────────────────────────────────────────────────────────────┘
```
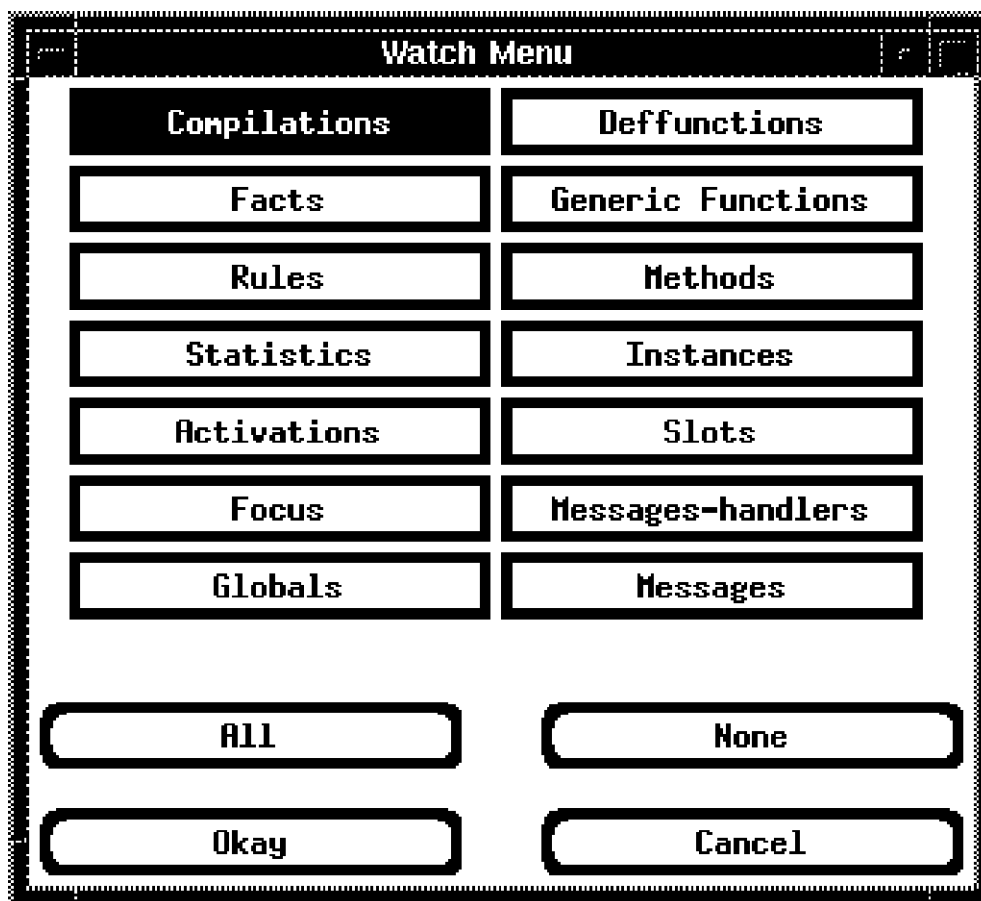
Watch items can be enabled or disabled by clicking its check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item check boxes. Pressing the **None** button unchecks all of the watch item check boxes. Clicking the **Okay** button exits the dialog and changes the current watch settings to those shown in the dialog. Clicking the **Cancel** button exits the dialog, but retains the original settings of the watch items before the dialog was entered.

This command is available when CLIPS is executing.

4.3.5 Options...

This command displays a dialog box which allows the user to set various CLIPS execution options.

The **Salience Evaluation** pop-up menu allows the current salience evaluation behavior to be changed (to either when-defined, when-activated, or every-cycle). The **Strategy** pop-up menu allows the current conflict resolution strategy to be changed (to either depth, breadth, lex, mea, complexity, simplicity, or random). The **Static Constraint Checking**, **Dynamic Constraint Checking**, **Reset Global Variables**, **Sequence Expansion Operator Recognition**, **Incremental Reset**, **Auto-Float Dividend**, and **Fact Duplication** check boxes are used to enable and disable the named CLIPS option. Enabled options have a check in their check box while disabled items have none. The **Incremental Reset** option cannot be changed if any defrules are currently defined. Clicking the **OK** button exits the dialog and changes the CLIPS execution options settings to those shown in the dialog. Clicking the **Cancel** button exits the dialog, but retains the original settings of the CLIPS execution options before the dialog was entered.

This command is available when CLIPS is executing.

4.3.6 Clear CLIPS (^K)

This command is equivalent to the CLIPS command (clear). When this command is chosen, the CLIPS command (clear) will be echoed to the main window and executed. A dialog box will appear issuing a warning and providing an opportunity to cancel this command. This command is not available when CLIPS is executing.

4.3.7 Clear Window (^N)

This command clears the dialog window. This command is not available when CLIPS is executing.

4.4 THE BROWSE MENU

```
┌─────────────────────┐
│      Browse         │
└─────────────────────┘
┌─────────────────────┐
│ Module...           │
│  Defrule Manager... │
│  Deffacts Manager...│
│  Deftemplates Manager...│
│  Deffunction Manager...│
│  Defglobal Manager...│
│  Defgeneric Manager...│
│  Defclass Manager...│
│  Definstances Manager...│
│  Agenda Manager...  │
└─────────────────────┘
```
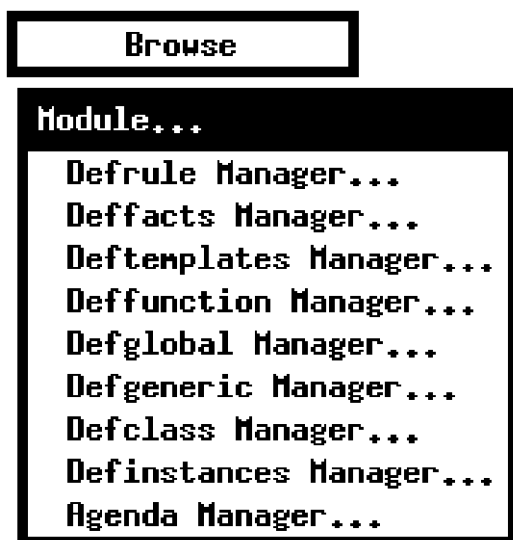
4.4.1 Module Menu

This menu displays a dialog box which contains the list of the defmodules currently defined in the CLIPS environment. A current module is hilighted. Selecting a module from the list changes the current module to the selected item.
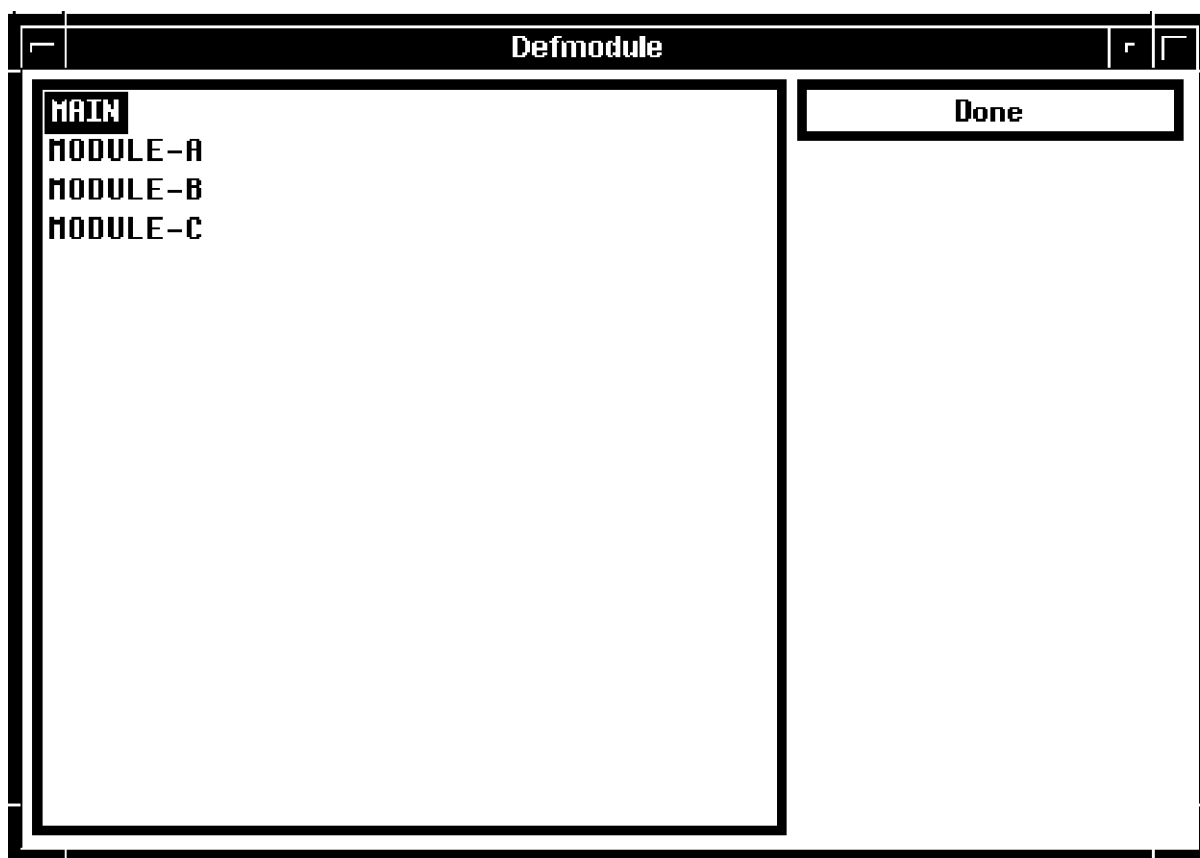
When finish selecting the module click the **Done** button to remove the dialog.

4.4.2 Defrule Manager...

This command displays a dialog box which allows the defrules in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defrule selected from the list of defrules currently in the knowledge base.

The **Remove** button will remove the currently selected defrule. Pressing this button is equivalent to issuing an **undefrule** command. The **Refresh** button will refresh the currently selected defrule. Pressing this button is equivalent to a **refresh** command. The **Matches** command will list the matches for the currently selected defrule. Pressing this button is equivalent to a **matches** command. The **Pprint** button will pretty print the currently selected defrule. Pressing this button is equivalent to a **ppdefrule** command. Both the **Matches** and **Pprint** buttons echo the equivalent CLIPS command and resulting output to the dialog window.

The **Breakpoint** check box is used to enable or disable a breakpoint on the currently selected defrule. Rules with a breakpoint set have a check in their check box while rules without a breakpoint have none. The **Watch Activations** check box is used to enable or disable the messages that are printed when a rule is activated or deactivated. If the check box for a rule is checked, then activation/deactivation messages for the rule will be printed. The **Watch Firings** check box is used to enable or disable the messages that are printed when a rule is executed. If the check box for a rule is checked, then execution messages for the rule will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defrule Manager...** command is not available when CLIPS is executing.

4.4.3 Deffacts Manager...

This command displays a dialog box which allows the deffacts in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffacts selected from the list of deffacts currently in the knowledge base.

The **Remove** button will remove the currently selected deffacts. Pressing this button is equivalent to an **undeffacts** command The **Pprint** button will pretty print the currently selected deffacts. Pressing this button is equivalent to a **ppdeffacts** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deffacts Manager...** command is not available when CLIPS is executing.

4.4.4 Deftemplate Manager...

This command displays a dialog box which allows the deftemplates in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deftemplate selected from the list of deftemplates currently in the knowledge base.

```
┌──────────────── Deftemplate Manager - 13 Items ────────────────┐
│  ┌─────────────────────────────────┐  ┌──────────────────────┐ │
│  │ deftemplate-A                   │  │       Remove         │ │
│  │ deftemplate-B                   │  └──────────────────────┘ │
│  │ deftemplate-C                   │  ┌──────────────────────┐ │
│  │ deftemplate-D                   │  │       Pprint         │ │
│  │ deftemplate-E                   │  └──────────────────────┘ │
│  │ deftemplate-F                   │  ┌─┐  Watch              │ │
│  │ deftemplate-G                   │  └─┘                     │ │
│  │ deftemplate-H                   │  ┌──────────────────────┐ │
│  │ deftemplate-I                   │  │       Done           │ │
│  │ deftemplate-J                   │  └──────────────────────┘ │
│  │ deftemplate-K                   │                           │
│  │ deftemplate-L                   │                           │
│  │ deftemplate-M                   │                           │
│  │                                 │                           │
│  └─────────────────────────────────┘                          │
└────────────────────────────────────────────────────────────────┘
```

The **Remove** button will remove the currently selected deftemplate. Pressing this button is equivalent to an **undeftemplate** command. The **Pprint** button will pretty print the currently selected deftemplate. Pressing this button is equivalent to a **ppdeftemplate** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed whenever a fact is asserted or retracted. If the check box for a deftemplate is checked, then assert/retract messages for facts associated with the deftemplate will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deftemplate Manager...** command is not available when CLIPS is executing.

4.4.5 Deffunction Manager...

This command displays a dialog box which allows the deffunctions in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffunction selected from the list of deffunctions currently in the knowledge base.

The **Remove** button will remove the currently selected deffunction. Pressing this button is equivalent to an **undeffunction** command. The **Pprint** button will pretty print the currently selected deffunction. Pressing this button is equivalent to a **ppdeffunction** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a deffunction begins and ends execution. If the check box for a deffunction is checked, then execution messages for the deffunction will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Deffunction Manager...** command is not available when CLIPS is executing.

4.4.6 Defglobal Manager

This command displays a dialog box which allows the defglobals in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defglobal selected from the list of defglobals currently in the knowledge base. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defglobal Manager...** command is not available when CLIPS is executing.

```
╔══════════════════ Defglobal Manager - 13 Items ══════════════════╗
║  ┌──────────────────────────┐   ┌─────────────────┐             ║
║  │ defglobal-A              │   │     Remove      │             ║
║  │ defglobal-B              │   └─────────────────┘             ║
║  │ defglobal-CC             │   ┌─────────────────┐             ║
║  │ defglobal-D              │   │     Pprint      │             ║
║  │ defglobal-E              │   └─────────────────┘             ║
║  │ defglobal-F              │   ┌─────────────────┐             ║
║  │ defglobal-G              │   │      Done       │             ║
║  │ defglobal-H              │   └─────────────────┘             ║
║  │ defglobal-I              │                                   ║
║  │ defglobal-J              │                                   ║
║  │ defglobal-K              │                                   ║
║  │ defglobal-L              │                                   ║
║  │ defglobal-M              │                                   ║
║  └──────────────────────────┘                                   ║
╚══════════════════════════════════════════════════════════════════╝
```

The **Remove** button will remove the currently selected defglobal. Pressing this button is equivalent to an **undefglobal** command The **Pprint** button will pretty print the currently selected deffacts. Pressing this button is equivalent to a **ppdefglobal** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defglobal Manager...** command is not available when CLIPS is executing.

4.4.7 Defgeneric Manager

This command displays a dialog box which allows the defgenerics in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defgeneric selected from the list of defgenerics currently in the knowledge base.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defgeneric Manager...** command is not available when CLIPS is executing.

The **Remove** button will remove the currently selected defgeneric. Pressing this button is equivalent to an **undefgeneric** command. The **Pprint** button will pretty print the currently selected defgeneric. Pressing this button is equivalent to a **ppdefgeneric** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a generic function begins and ends execution. If the check box for a deffunction is checked, then execution messages for the generic function will be printed.

The **Methods...** button displays a dialog box allowing the defmethods for the currently selected defgeneric to be browsed. The dialog box that appears in response to this button allows an operation to be performed on a defmethod selected from the list of defmethods currently in the knowledge base.
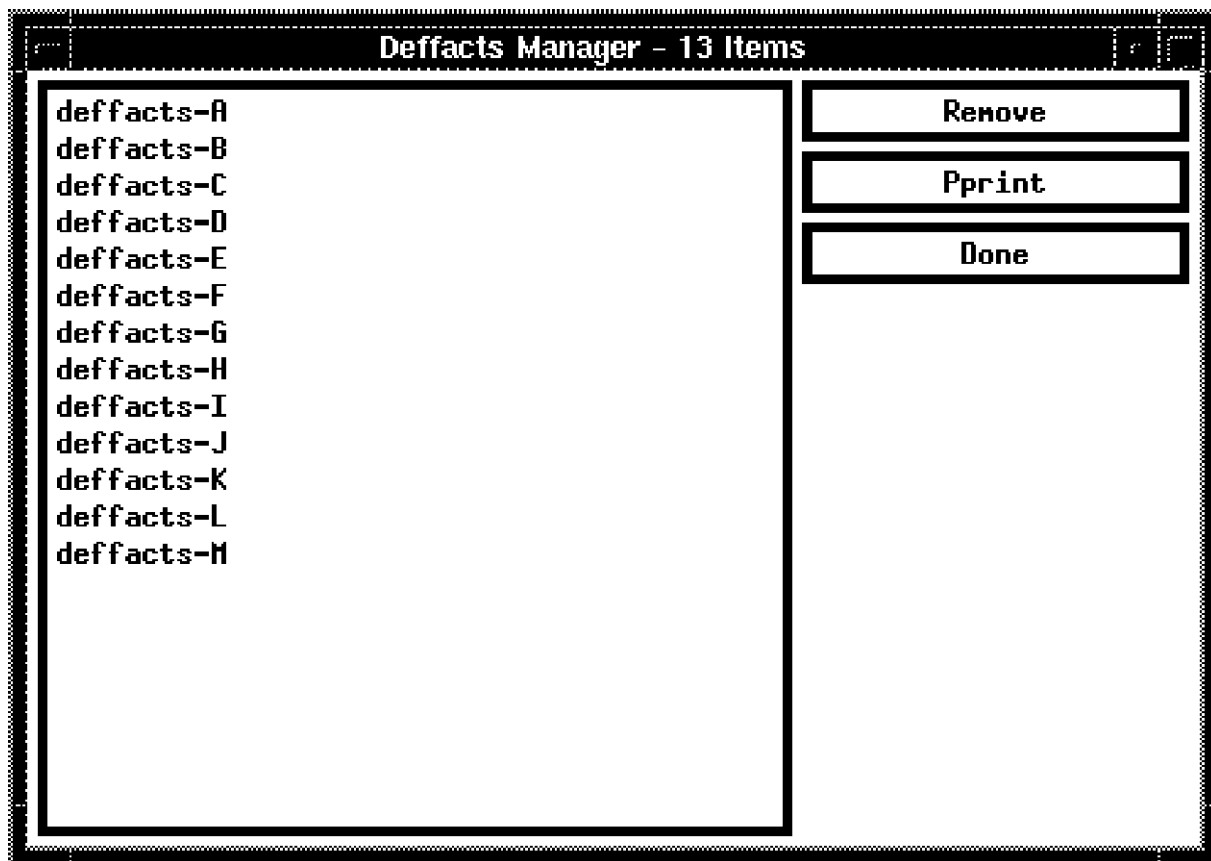
```
┌──────────────────────────────────────────────────────────────────┐
│  ┌──┐      defgeneric-A Defmethod Manager - 11 Items      ┌─┐┌──┐  │
│  └──┘                                                     └─┘└──┘  │
│  ┌─────────────────────────────────────┐  ┌────────────────────┐  │
│  │ 1  (FLOAT)                          │  │       Remove       │  │
│  │ 10 (STRING NUMBER)                  │  └────────────────────┘  │
│  │ 11 (SYMBOL NUMBER)                  │  ┌────────────────────┐  │
│  │ 2  (INTEGER)                        │  │       Pprint       │  │
│  │ 3  (NUMBER)                         │  └────────────────────┘  │
│  │ 4  (SYMBOL)                         │  ┌──┐ ┌──────────┐        │
│  │ 5  (STRING)                         │  │  │ │  Watch   │        │
│  │ 6  (STRING INTEGER)                 │  └──┘ └──────────┘        │
│  │ 7  (SYMBOL INTEGER)                 │  ┌────────────────────┐  │
│  │ 8  (STRING FLOAT)                   │  │        Done        │  │
│  │ 9  (SYMBOL FLOAT)                   │  └────────────────────┘  │
│  │                                     │                          │
│  │                                     │                          │
│  │                                     │                          │
│  │                                     │                          │
│  │                                     │                          │
│  │                                     │                          │
│  │                                     │                          │
│  └─────────────────────────────────────┘                          │
└──────────────────────────────────────────────────────────────────┘
```

The **Remove** button will remove the currently selected defmethod. Pressing this button is equivalent to an **undefmethod** command. The **Pprint** button will pretty print the currently selected defmethod. Pressing this button is equivalent to a **ppdefmethod** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a specific method of a generic function begins and ends execution. If the check box for a defmethod is checked, then execution messages for the method will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to return control to the **Defgeneric Manager** dialog box.
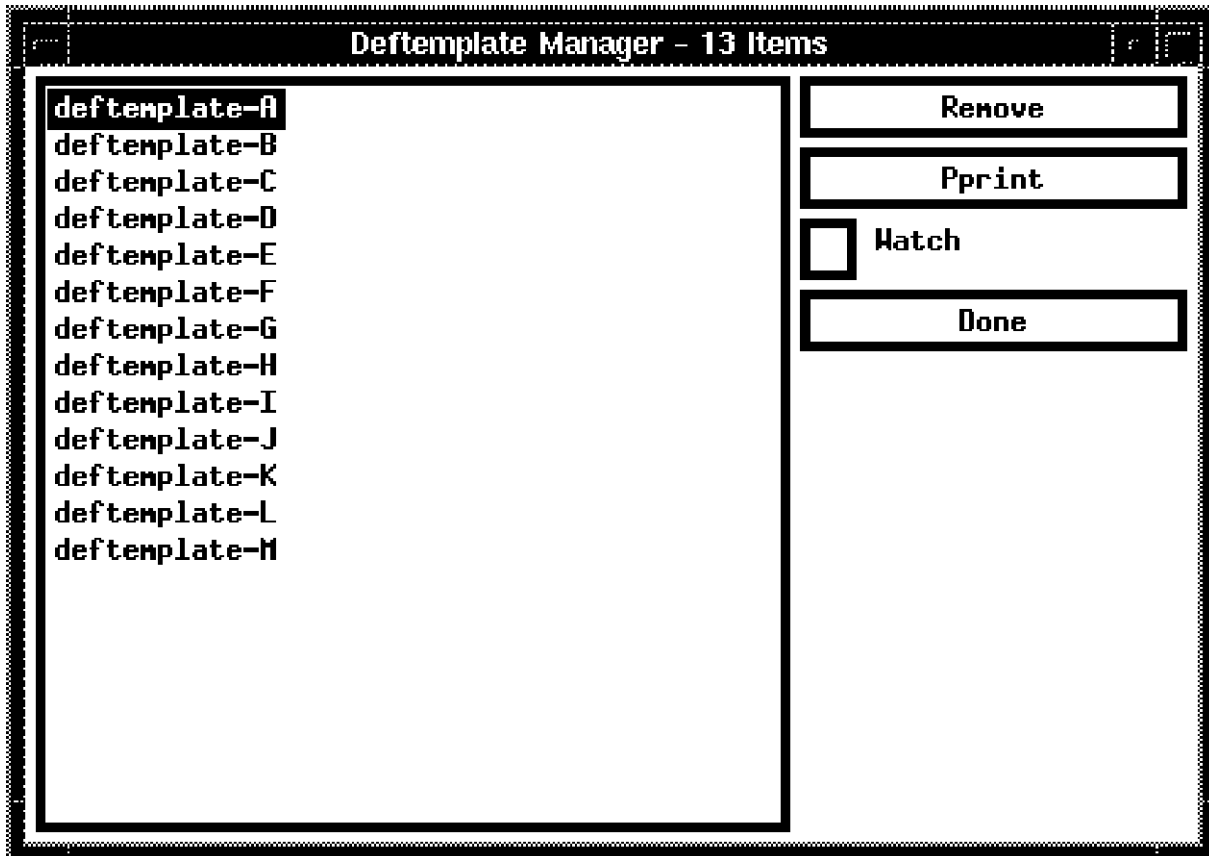
4.4.8 Defclass Manager...

This command displays a dialog box which allows the defclasses in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defclass selected 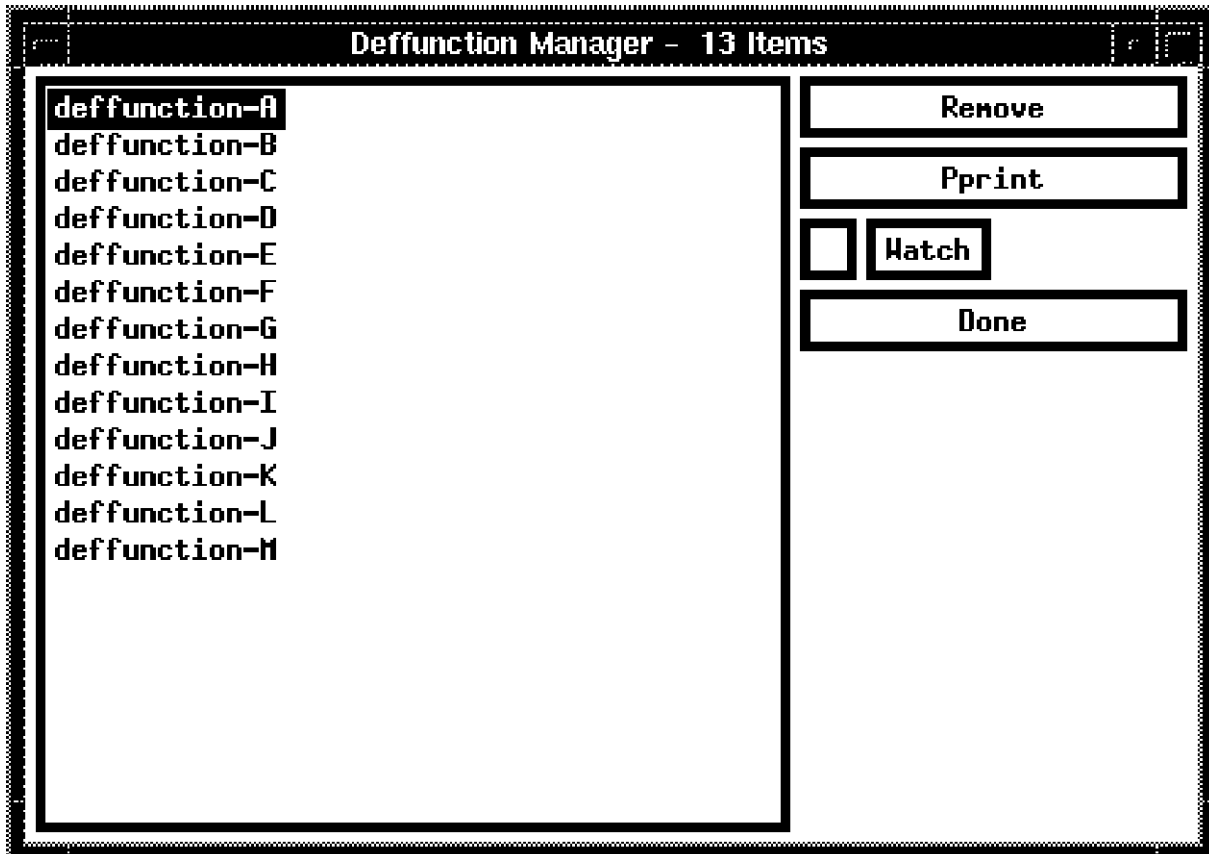from the lis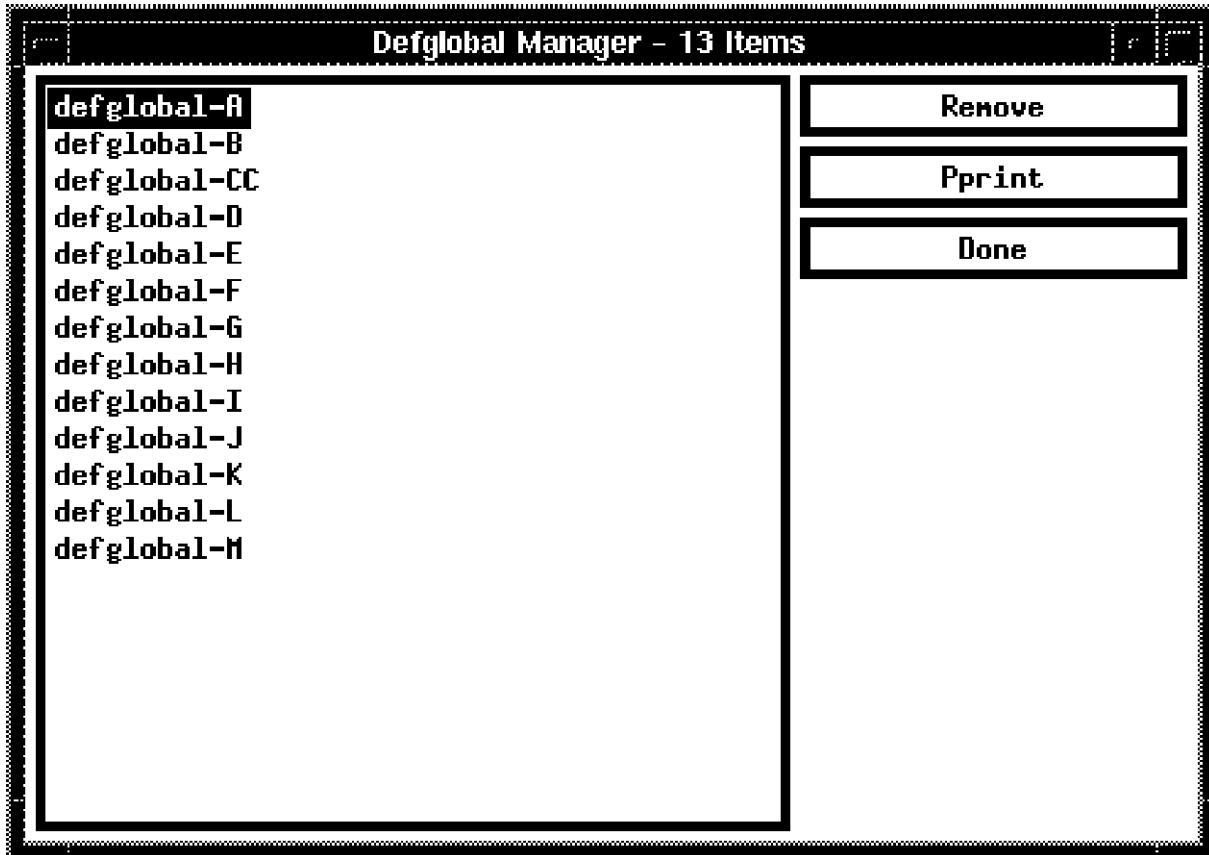t of defclasses currently in the knowledge base. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defclass Manager...** command is not available when CLIPS is executing.

The **Remove** button will remove the currently selected defclass. Pressing this button is equivalent to an **undefclass** command. The **Describe** button will describe the currently selected defclass. Pressing this button is equivalent to a **describe-class** command. The **Browse** command displays the class hierarchy for the currently selected defclass. Pressing this button is equivalent to a **browse-class** command. The **Pprint** button will pretty print the currently selected defclass. Pressing this button is equivalent to a **ppdefclass** command. The **Describe**, **Browse**, and **Pprint** buttons echo the equivalent CLIPS command and resulting output to the dialog window.

The **Watch Instances** check box is used to enable or disable the messages that are printed when an instance is created or deleted. If the check box for a defclass is checked, then messages will be printed whenever an instance of that class is created or deleted. The **Watch Slots** check box is used to enable or disable the messages that are printed when an instance's slot values are changed. If the check box for a defclass is checked, then messages will be printed whenever an instance of that class has a slot value changed.

The **Message Handlers...** button displays a dialog box allowing the message handlers for the currently selected class to be browsed. The dialog box that appears in response to this button allows an operation to be performed on a defmessage-handler selected from the list of defmessage-handlers currently in the knowledge base.

```
┌─────────────────────────────────────────────────────────────────────────┐
│ ▓▓    Defclass Manager - 18 Items                         ▓ ▓ □          │
│  ┌──────────────────────────────────────────────────────────────────┐   │
│  │ ▓    DEFCLASS-A Defmessage-Handler Manager - 12 Items      ▓ ▓ █  │   │
│  │  ┌────────────────────────────────────┐  ┌───────────────────┐    │   │
│  │  │ defmessage-handler-A primary       │  │      Remove       │    │   │
│  │  │ defmessage-handler-B primary       │  └───────────────────┘    │   │
│  │  │ defmessage-handler-C primary       │  ┌───────────────────┐    │   │
│  │  │ defmessage-handler-D primary       │  │      Pprint       │    │   │
│  │  │ defmessage-handler-E primary       │  └───────────────────┘    │   │
│  │  │ defmessage-handler-F primary       │  ┌──┐ ┌─────────┐         │   │
│  │  │ defmessage-handler-G primary       │  │  │ │  Watch  │         │   │
│  │  │ defmessage-handler-H primary       │  └──┘ └─────────┘         │   │
│  │  │ defmessage-handler-I primary       │  ┌───────────────────┐    │   │
│  │  │ defmessage-handler-J primary       │  │       Done        │    │   │
│  │  │ defmessage-handler-K primary       │  └───────────────────┘    │   │
│  │  │ defmessage-handler-L primary       │                           │   │
│  │  │              .                     │                           │   │
│  │  │                                    │                           │   │
│  │  │                                    │                           │   │
│  │  │                                    │                           │   │
│  │  │                                    │                           │   │
│  │  │                                    │                           │   │
│  │  │                                    │                           │   │
│  │  └────────────────────────────────────┘                           │   │
│  └──────────────────────────────────────────────────────────────────┘   │
└─────────────────────────────────────────────────────────────────────────┘
```

The **Remove** button will remove the currently selected defmessage-handler. Pressing this button is equivalent to an **undefmessage-handler** command. The **Pprint** button will pretty print the currently selected defmessage-handler. Pressing this button is equivalent to a **ppdefmessage-handler** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

The **Watch** check box is used to enable or disable the messages that are printed when a message-handler begins and ends execution. If the check box for a defmessage-handler is checked, then execution messages for the message-handler will be printed.

Any number of operations can be executed from this dialog. When finished, click the **Done** button to return control to the **Defclass Manager** dialog box.

4.4.9 Definstances Manager...

This command displays a dialog box which allows the definstances in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a definstances selected from the list of definstances currently in the knowledge base.
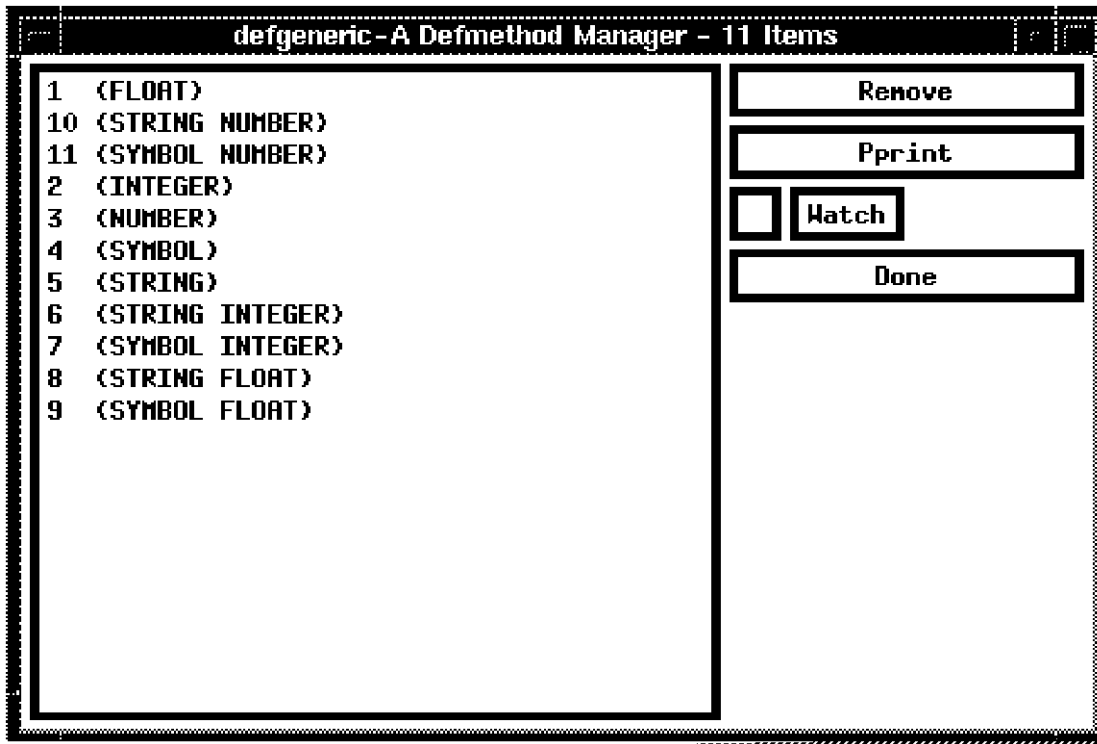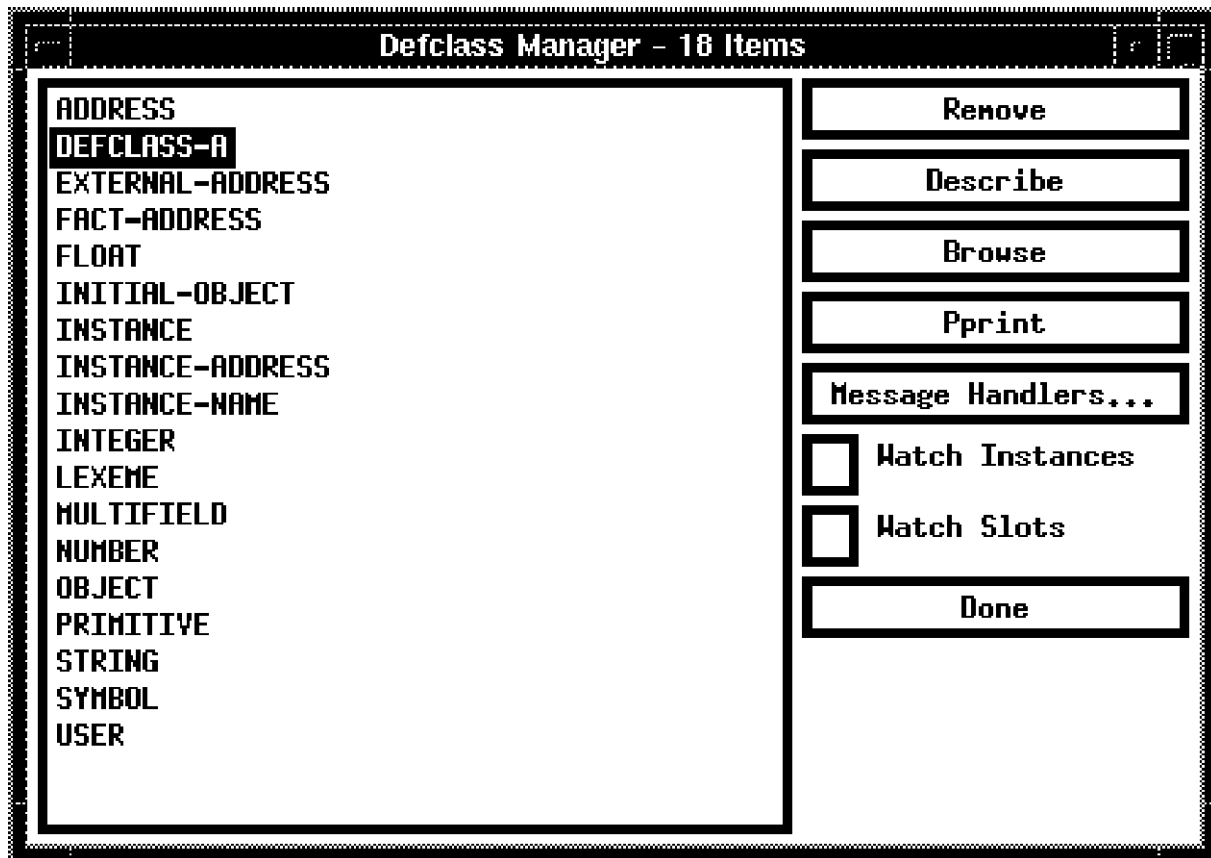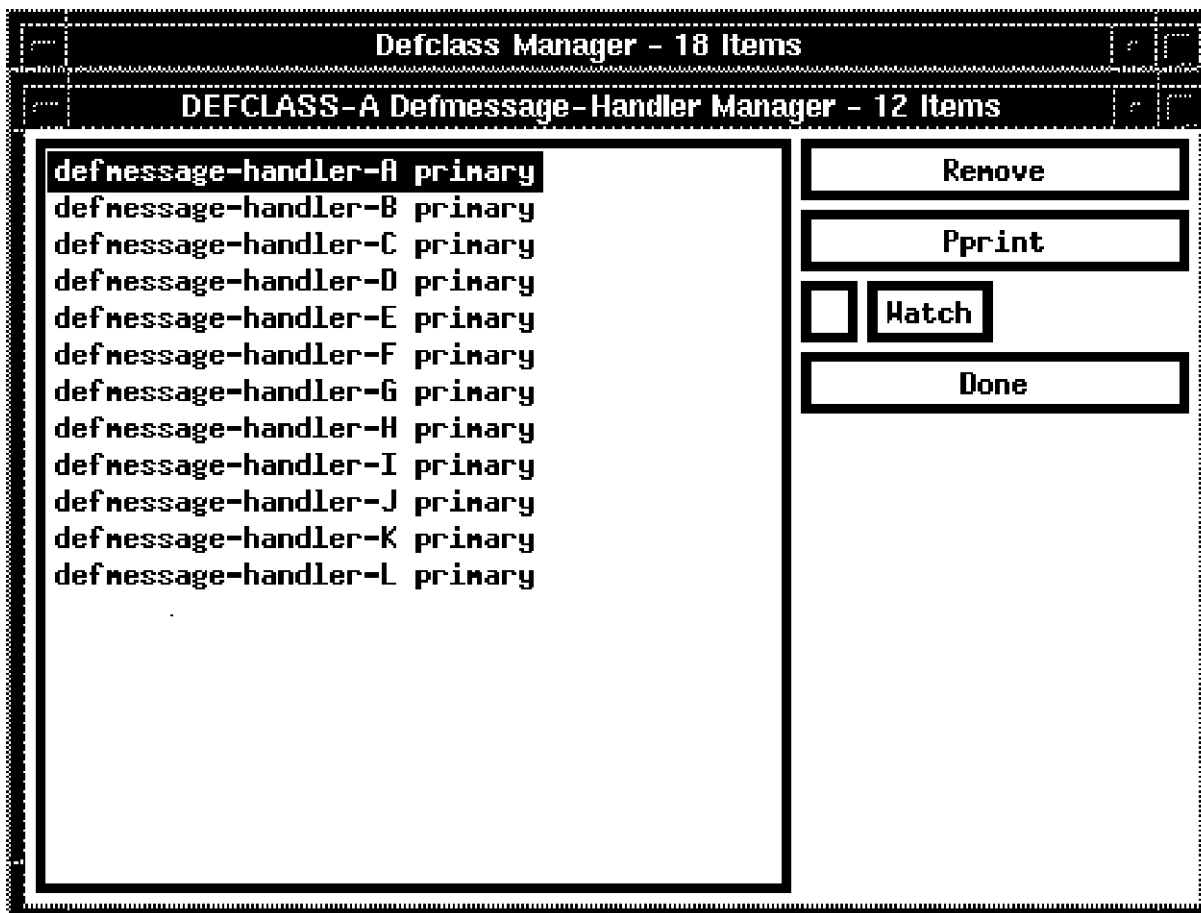
```
┌─────────────────────────────────────────────────────────┐
│ ▪    ▪         Definstance Manager – 13 Items      ▪  ▪  │
│ ┌─────────────────────────────────┐ ┌─────────────────┐ │
│ │ definstances-A                  │ │     Remove      │ │
│ │ definstances-B                  │ └─────────────────┘ │
│ │ definstances-C                  │ ┌─────────────────┐ │
│ │ definstances-D                  │ │     Pprint      │ │
│ │ definstances-E                  │ └─────────────────┘ │
│ │ definstances-F                  │ ┌─────────────────┐ │
│ │ definstances-G                  │ │      Done       │ │
│ │ definstances-H                  │ └─────────────────┘ │
│ │ definstances-J                  │                     │
│ │ definstances-K                  │                     │
│ │ definstances-L                  │                     │
│ │ definstances-M                  │                     │
│ │ initial-object                  │                     │
│ │                                 │                     │
│ │                                 │                     │
│ │                                 │                     │
│ │                                 │                     │
│ │                                 │                     │
│ │                                 │                     │
│ │                                 │                     │
│ └─────────────────────────────────┘                     │
└─────────────────────────────────────────────────────────┘
```

The **Remove** button will remove the currently selected definstances. Pressing this button is equivalent to an **undefinstances** command. The **Pprint** button will pretty print the currently selected definstances. Pressing this button is equivalent to a **ppdefinstances** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window. Any number of operations can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Definstances Manager...** command is not available when CLIPS is executing.

4.4.10 Agenda Manager...

This command displays a dialog box which allows the activations on the agenda to be examined. The dialog box that appears in response to this command allows any activation on the agenda to be removed or fired.

```
┌─────────────────────────────────────────────────────────────────────┐
│  ┌──┐              Agenda Manager - 13 Items              ┌─┐┌──┐    │
│  └──┘ ·················································     └─┘└──┘    │
│  ┌────────────────────────────────────────┐  ┌─────────────────────┐ │
│  │ 0      activation-13: f-3,f-2,f-1       │  │       Remove        │ │
│  │ 0      activation-4: f-3,f-2            │  └─────────────────────┘ │
│  │ 0      activation-5: f-3,f-1            │  ┌─────────────────────┐ │
│  │ 0      activation-7: f-3,f-1            │  │        Fire         │ │
│  │ 0      activation-1: f-3                │  └─────────────────────┘ │
│  │ 0      activation-10: f-3               │  ┌─────────────────────┐ │
│  │ 0      activation-9: f-2,f-3            │  │        Done         │ │
│  │ 0      activation-11: f-1,f-3,f-2       │  └─────────────────────┘ │
│  │ 0      activation-12: f-2,f-1,f-3       │                          │
│  │ 0      activation-6: f-2,f-1            │                          │
│  │ 0      activation-2: f-2                │                          │
│  │ 0      activation-8: f-1,f-2            │                          │
│  │ 0      activation-3: f-1                │                          │
│  │                                         │                          │
│  │                                         │                          │
│  │                                         │                          │
│  └─────────────────────────────────────────┘                         │
└─────────────────────────────────────────────────────────────────────┘
```
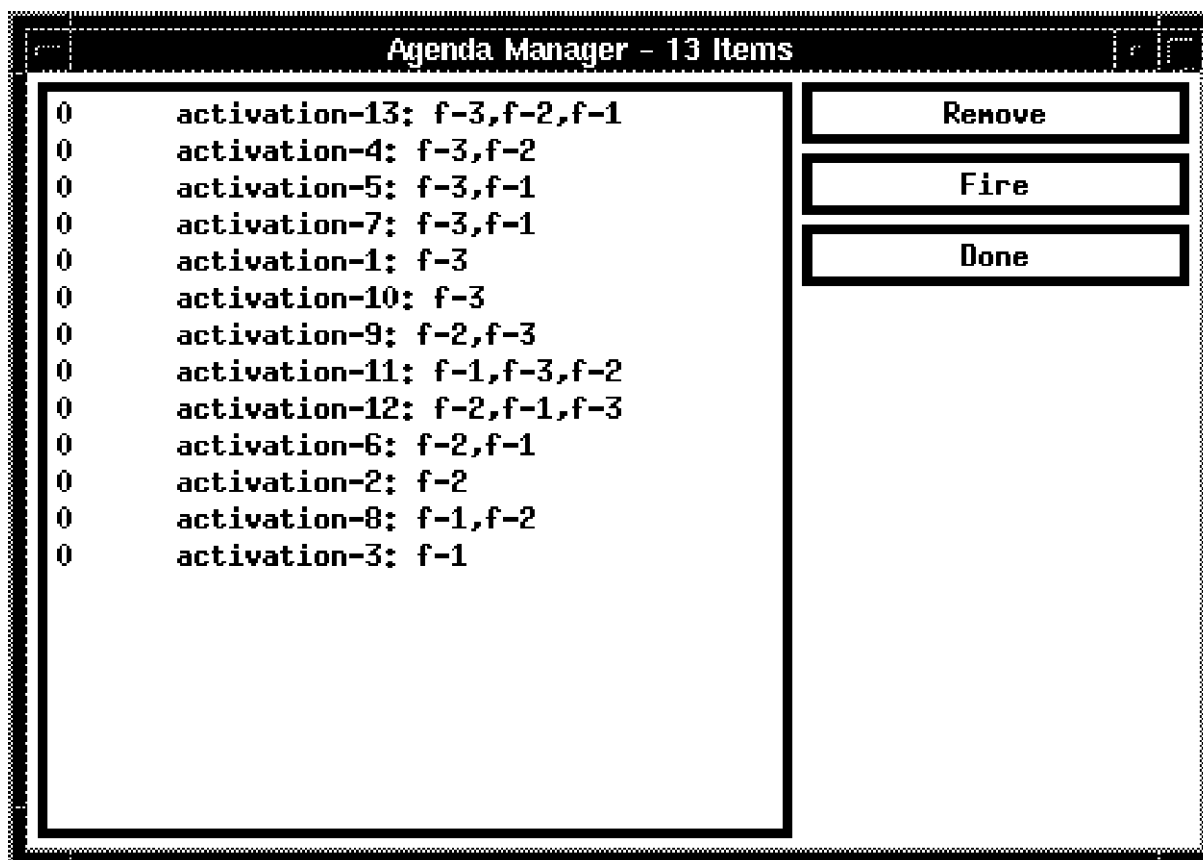
The **Remove** button will remove the currently selected activation from the agenda. The **Fire** button will place the currently selected activation at the top of the agenda (regardless of its salience). The dialog is then exited and the CLIPS command (run 1) will then be echoed to the **Dialog Window**, causing the activation to fire. Any number of remove operations (except the **Fire** button) can be executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Agenda Manager...** command is not available when CLIPS is executing.
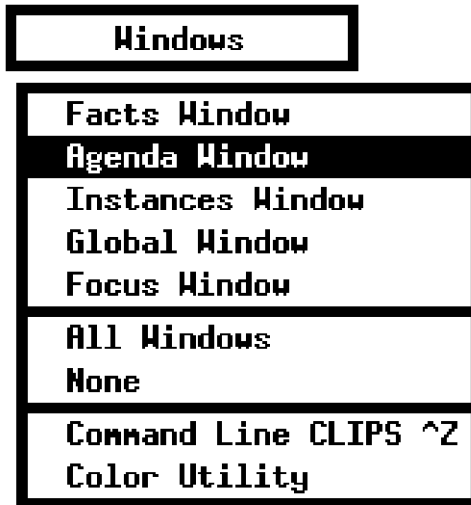
4.5 THE WINDOW MENU

```
┌─────────────────────────┐
│        Windows          │
├─────────────────────────┤
│    Facts Window         │
│    Agenda Window        │
│    Instances Window     │
│    Global Window        │
│    Focus Window         │
├─────────────────────────┤
│    All Windows          │
│    None                 │
├─────────────────────────┤
│    Command Line CLIPS ^Z│
│    Color Utility        │
└─────────────────────────┘
```

4.5.1 Facts Window

This command creates or deletes the **Facts Window**. The **Facts Window** displays the list of facts in the fact-list. An X Window logo is placed by the window name to indicated that it exists. This command is available when CLIPS is executing.

4.5.2 Agenda Window

This command creates or deletes the **Agenda Window**. The **Agenda Window** displays the list of activations currently on the agenda. An X Window logo is placed by the window name to indicated that it exists. This command is available when CLIPS is executing.

4.5.3 Instances Window

This command creates or deletes the **Instances Window**. The **Instances Window** displays the list of existing instances and their slot values. A check mark is placed by the window name to indicated that it exists. This command is available when CLIPS is executing.

4.5.4 Globals Window

This command creates or deletes the **Globals Window**. The **Globals Window** displays the list of global variables and their current values. A check mark is placed by the window name to indicated that it exists. This command is available when CLIPS is executing.

4.5.5 Focus Window

This command selects the **Focus Window** and brings it to the front. If the **Focus Window** does not exist, it will be created. The **Focus Window** displays the current focus stack. A check mark is placed by the window name to indicated that it is the frontmost window. This command is available when CLIPS is executing.

4.5.6 All

This command creates the **Dialog Window**, **Facts Window**, **Instances Window**, and **Globals Window** if they do not already exist. This command is available when CLIPS is executing.

4.5.7 None

This command deletes the **Dialog Window**, **Facts Window**, **Instances Window**, and **Globals Window** if they exist. This command is available when CLIPS is executing.

4.5.8 Command Line CLIPS (^Z)

This command creates a *xterm* window and starts a command line version of CLIPS. This window will still exist after the **xclips** interface has ended. This command is available when CLIPS is executing.

4.5.9 Color Utility

This command will execute the color utility for the **xclips** interface. To use the new colors the Quit command with the Restart (see section 4.2.9) option must be chosen.

4.5.9.1 How To Use The Color Utility

To change the colors of a window in the X Window interface for CLIPS, the user must:
1) Select the appropriate button for the window at the bottom of the color utility.
2) Select the item that needs to be changed at the top of the color utility.
3) Select the color that is listed on the left of the color utility.

The instant visual changes on the window will help the user to decide which colors he/she likes best. There are 14 different buttons at the bottom of the color utility that the user can select. They are listed as follows:

Dialog Window: The main interactive window that accepts input from users and prints out output from CLIPS.

Facts Window: The fact window that contains all the facts in the fact base.

Instance Window: The instance window that contains all the instances.

Globals: The global window that contains all the global variables and their values.

Agenda Window: The agenda window that contains all the activated rules.

Focus Window: The focus window that contains all the modules in the stack.

Button Box: The box that contains all the top level menu buttons. Only the background color can be changed here.

Menu Buttons: All the top level menu buttons.

Pulldown Menus: All the pulldown menus.

Managers: All the defconstruct manager windows (under the Browse menu).

Managers Button: There are regular buttons and a **Cancel** button, which the user can modify their colors, in a manager window. Their foreground color and their border color will always be the same.

Confirmation: The confirmation dialog box.

File: The file selection window which popups when the name of a file is needed.

Watch/Options: The windows that allow the user to select the watched items and the options (created using the **Watch** and **Options** menu items).

Wtch/Opt Bttns : Toggle buttons and command buttons in the **Watch** or **Options** menu items.

On the top of the color utility there are five buttons the user can select, **Foreground, Background, Border, Default, Quit.** When the foreground and border colors are not allowed to change, the label of the **Foreground** and **Border** buttons will change to **N/A** or to something else that appropriate for the current selected button at the bottom of the color utility. Clicking on Default will give the user the default colors. When the user is happy with his/her selections and wants to save them, she/he can click on **Save and Quit** under the **Quit** menu**.** Selecting **Save and Quit** instructs the color utility to save the current color resource setting in the file **Xclips** which must reside in the user's home directory.

4.6 THE EDITOR

This section provides a brief summary for the editor interface. The editor is an instance of the Athena asciiText widget, which comes with X and is similar to the *emacs* editor.

4.6.1 File

```
┌─────────────────────────┐
│         File            │
├─────────────────────────┤
│ Save                    │
│ Save As                 │
│ Revert                  │
├─────────────────────────┤
│ Load Selection          │
│ Batch Selection         │
│ Load Buffer             │
├─────────────────────────┤
│ Exit                    │
└─────────────────────────┘
```
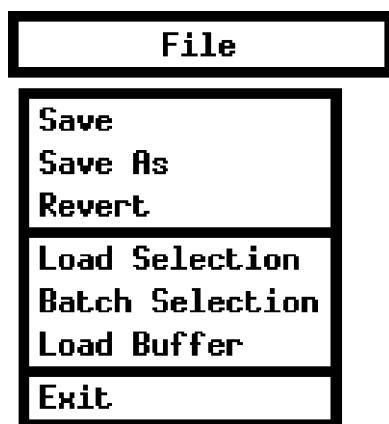
4.6.1.1 Save

This command displays a dialog box allowing the user to overwrite the old file with the edited file or cancel the save command. This command is not available when CLIPS is executing.

4.6.1.2 Save As

This command allows the edit window to be saved under a new name. A dialog box will appear to prompt for the new file name. The name of the editing window will be changed to the new file name. This command is not available when CLIPS is executing.

4.6.1.3 Revert

This command restores the edit window to the last saved version of the file in the buffer. Any changes made since the file was last saved will be discarded. This command is not available when CLIPS is executing.

4.6.1.4 Load Selection

This command loads the current selection in the active edit window into the CLIPS knowledge base. Standard error detection and recovery routines used to load constructs from a file are also used when loading a selection (i.e., if a construct has an error in it, the rest of the construct will be skipped over until another construct to be loaded is found). This command is not available when CLIPS is executing.

4.6.1.5 Batch Selection

This command treats the current selection in the active edit window as if it were a batch file and executes it as a series of commands. Standard error detection and recovery routines used to load construct from a file are *not* used when batching a selection (i.e., if a construct has an error in it, a number of ancillary errors may be generated by subsequent parts of the same construct following the error). This command is not available when CLIPS is executing.
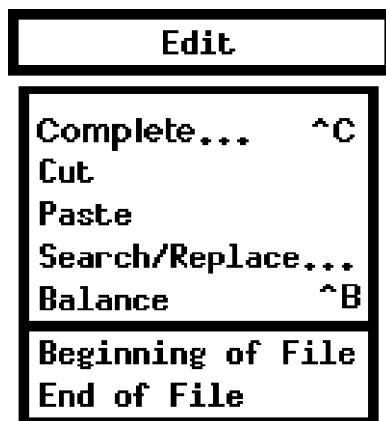
4.6.1.6 Load Buffer

This command loads the contents of the active edit window into the CLIPS knowledge base. It is equivalent to selecting the entire buffer and executing a **Load Selection** command. This command is not available when CLIPS is executing.
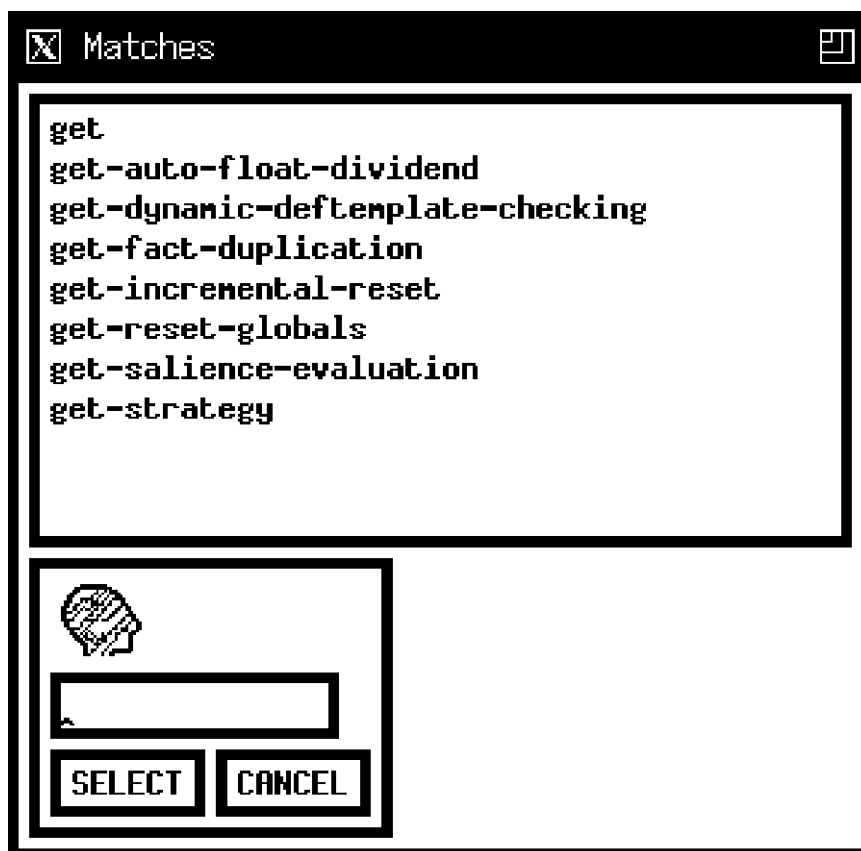
4.6.1.7 Exit

This command exits the editor. This command is not available when CLIPS is executing.

4.6.2 Edit

```
┌─────────────────────┐
│        Edit         │
├─────────────────────┤
│ Complete...      ^C  │
│ Cut                 │
│ Paste               │
│ Search/Replace...   │
│ Balance          ^B  │
├─────────────────────┤
│ Beginning of File   │
│ End of File         │
└─────────────────────┘
```

4.6.2.1 Complete... (^C)

This command "completes" the symbol currently being entered in the active edit window. The symbol being completed is the current selection, or if no selection exists, the symbol to the left of the current insertion point. If no possible completions exist, a beep is sounded. If only one possible completion exists (e.g. "deftemplat" for "deftemplate"), then the symbol is automatically completed. If more than one completion exists, a dialog is displayed showing all possible completions. The following dialog shows the possible completions for the symbol "get":

Click the **Select** button to complete the symbol with the current selection. Click the **Cancel** button to be terminate the dialog without any command completion. Note that symbols defined by the user (as part of a construct or a CLIPS data structure such as a fact or instance) will also be listed for command completion. This command is not available when CLIPS is executing.

4.6.2.2 Cut

This command removes selected text in the active edit window and places it in the Clipboard. This command is not available when CLIPS is executing.

4.6.2.3 Paste

This command copies the contents of the Clipboard to the selection point in the edit window. This command is not available when CLIPS is executing.

4.6.2.4 Search/Replace... (^S/^R)

This command displays the search/replace dialog allowing the user to search for a string or search for a string and replace it with another string. This command is not available when CLIPS is executing.

4.6.2.5 Balance (^B)

This command operates on the current selection in the active edit window by attempting to find the smallest selection containing the current selection which has balanced parentheses. Repeatedly using this command will select larger and larger selections of text until a balanced selection cannot be found. The balance command is a purely textual operation and can be confused by parentheses found in strings. The cursor has to be placed to the left of the parenthesis. This command is not available when CLIPS is executing.
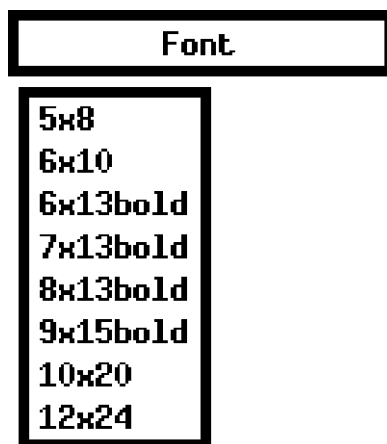
4.6.2.6 Beginning of File

This command will scroll the insertion point to the beginning of the editor file and then scroll the minimum amount necessary to make the new insert point location visible. This command is not available when CLIPS is executing.

4.6.2.7 End of File

This command will scroll the insertion point to the end of the editor file and then scroll the minimum amount necessary to make the new insert point location visible. This command is not available when CLIPS is executing.

4.6.3 Font

```
┌─────────────────────┐
│        Font         │
├─────────────────────┤
│ 5x8                 │
│ 6x10                │
│ 6x13bold            │
│ 7x13bold            │
│ 8x13bold            │
│ 9x15bold            │
│ 10x20               │
│ 12x24               │
└─────────────────────┘
```

4.6.3.1 5x8

This command will set the editor font to 5x8. This command is not available when CLIPS is executing.

4.6.3.2 6x10

This command will set the editor font to 6x10. This command is not available when CLIPS is executing.

4.6.3.3 6x13bold

This command will set the editor font to 6x13bold. This command is not available when CLIPS is executing.

4.6.3.4 7x13bold

This command will set the editor font to 7x13bold. This command is not available when CLIPS is executing.

4.6.3.5 8x13bold

This command will set the editor font to 8x13bold. This command is not available when CLIPS is executing.

4.6.3.6 9x15bold

This command will set the editor font to 9x15bold. This command is not available when CLIPS is executing.

4.6.3.7 10x20

This command will set the editor font to 10x20. This command is not available when CLIPS is executing.

4.6.3.8 12x24

This command will set the editor font to 12x20. This command is not available when CLIPS is executing.

4.6.4 Help

```
        Help
```

```
Key Bindings
```

4.6.4.1 Key Bindings

This command displays a list of all key bindings available for the editor. This command is not available when CLIPS is executing.

4.7 BUILDING THE CLIPS X WINDOW INTERFACE

The following steps describe how to create the X Window machine specific version of CLIPS. This version of the CLIPS X Window interface requires X Window system release 11R6.4 and has been tested on a Macintosh running MacOS X 10.2.6 with Tenon Xtools version 1.2. Note that some familiarity with building applications using C compilers in a UNIX environment is assumed. Section 2 of the *Advanced Programming Guide* (which describes in general how to install and tailor CLIPS) should also be read before attempting to build the X Window executables.

1) The following source files are needed to build the interface in addition to the source files for the standard CLIPS executable:

   xclips.h   xclipstext.h   xedit.h   xmain.h
   xmenu.h   xmenu_exec.h   xmenu_file.h   xmenu_opt.h
   xmenu_watch.h   xmenu_wind.h   xsetup.h

   xclips.c   xclipstext.c   xedit.c   xmain.c
   xmenu.c   xmenu_exec.c   xmenu_file.c   xmenu_opt.c
   xmenu_watch.c   xmenu_wind.c

   Xclips

   The file **Xclips** is a resource file that must reside in the user's home directory in order for the X Window interface to appear in color. If this file cannot be found, the X Window interface will appear in black and white.

2) The compiler directive flags UNIX_7 and WINDOW_INTERFACE in the file **setup.h** should be set to 1.

3) The X Window system interface version of CLIPS can be created by compiling all the standard CLIPS .c files and the .c files listed in step 1. The file **xmain.c** replaces the file **main.c** used in the standard CLIPS (and thus **main.c** should not be compiled). The resulting object files should then be linked with the following libraries: Xaw, Xmu, Xt, Xext, X11, m. An example UNIX command for compiling **xclips** is shown following:

   ```
   gcc -o xclips *.c -lXaw -lXmu -lXt -lXext -lX11 -lm
   ```

Alternately, a makefile named "makefile.x" is included with the X Windows source. This file can be renamed "makefile" and used with the make program.

4) The color utility is a stand alone application. The source code files for this application are **color.c**, **default.h**, and **colors.h.** The executable version of the color utility can be generated by compiling **color.c** and linking it with the following libraries: Xaw, Xmu, Xt, Xext, X11. An example UNIX command for compiling the color utility is shown following:

```
gcc -o color color.c -lXaw -lXmu -lXt -lXext -lX11
```

Appendix A - Update Release Notes

The following sections denote the changes and bug fixes for CLIPS versions 6.01, 6.02, 6.03, and 6.04, 6.05, and 6.10, 6.20, 6.21, 6.23, 6.24, and 6.30.

A.1 VERSION 6.30

- **Compiler Support** - The following compilers are now supported.

  - Xcode 3.0 for Mac OS X.

  - Microsoft Visual C++ 2008 Express.

- **Windows Interface Bug Fixes** - The following bugs were fixed by the 6.30 release:

  - The position and size of the CLIPSWin window is now properly save and restored when the executable is exited and started.

- **Windows Interface Changes** - The following enhancements were added by the 6.30 release:

  - Text in the dialog window can be selected and copied.

  - Inline editing is supported in the dialog window.

  - A command history is supported for the dialog window.

  - In the dialog window, whenever the caret is placed to the immediate right of a right parenthesis, the matching left parenthesis is momentarily highlighted.

  - The (clear-window) function can be used to programmatically clear the contents of the dialog window.

  - Text files opened in an editing window have their line endings automatically converted to the Windows line ending convention if they are not already using that convention.

A.2 VERSION 6.24

- **Compiler Support** - The following compilers are now supported.

  - Metrowerks CodeWarrior 9.6 for Mac OS X.

- Xcode 2.3 for Mac OS X.

- Microsoft Visual C++ .NET 2003 for Windows.

- **Windows Interface Bug Fixes** - The following bugs were fixed by the 6.24 release:

  - Artifacts left by the insertion caret in the dialog window during scrolling are fixed.

  - The resize cursor did not appear to indicate a window could be resized.

- **Windows Interface Changes** - The following enhancements were added by the 6.24 release:

  - The position and size of the application's MDI frame window is save and restored between invocations of the program.

A.3 VERSION 6.23

- **Compiler Support** - The following compilers are now supported.

  - Metrowerks CodeWarrior 9.4 for Mac OS X and Windows.

  - Xcode 1.2 for Mac OS X.

A.4 VERSION 6.21

- **Windows Interface Bug Fixes** - The following bugs were fixed by the 6.21 release:

  - An array index error associated with the module menu was fixed.

- **Macintosh Interface Bug Fixes** - The following bugs were fixed by the 6.21 release:

  - The status windows were not updating properly.

- **X Windows Interface Bug Fixes** - The following bugs were fixed by the 6.21 release:

  - Numerous compilation problems were fixed.

A.5 VERSION 6.20

- **Windows Interface Changes** - The following enhancements were added by the 6.20 release:

- CLIPS engine and editor applications combined into a single Windows 95/98/NT Multiple Document Interface application.

- **Macintosh Interface Changes** - The following enhancements were added by the 6.20 release:

  - MacOS X support.

A.6 VERSION 6.10

- **Windows Interface Bug Fixes** - The following bugs were fixed by the 6.10 release:

  - The Windows 95 CLIPS interface did not properly display strings containing two sequential carriage returns.

  - Refresh and scroll bar bugs occurred when running the Windows 95 CLIPS interface under Windows NT.

- **Macintosh Interface Changes** - The following enhancements were added by the 6.10 release:

  - DOS/Unix line breaks are automatically converted to MacOS line breaks when a file is opened using the **Open…** menu item from the **File** menu.

A.7 VERSION 6.05

- **Windows Editor Interface Bug Fixes** - The following bugs were fixed by the 6.05 release:

  - If **Quit Editor** was selected from the **File** Menu before changes were saved, the cancel button of the save dialog did not abort the quit.

- **X Windows Interface Bug Fixes** - The following bugs were fixed by the 6.05 release:

  - A crash bug could occur in file dialogs when changing directories.

- **Macintosh Interface Changes** - The following enhancements were added by the 6.05 release:

  - In an editing window, the left margin is padded with the same number of spaces as found in the left margin of the previous line when the return key is pressed.

- A mark is placed by edit buffer window in the **Windows** menu to indicate whether it have been changed.

- Home, end, page up, page down, left arrow, right arrow, up arrow, and down arrow keys work for the **Display Window** and edit windows.

- Added **Comment**, **Uncomment**, and **Select All** menu items to the Edit Menu.

- Window position, selection, scroll positions, and font are remembered for edit windows. The position of the **Display Window** and other status windows are remembered.

- When a right parenthesis is typed, the matching left parenthesis will be momentarily highlighted.

- Watch, execution, and other preferences are now stored in a preference file and restored the next time the application is launched.

- Edit windows now contain a constructs pop-up menu in the lower left corner containing a list of constructs found in the window text.

- The ⌘ = command key is now used for the **Replace** menu item.

- The ⌘ T command key is now used for the **Replace & Find Again** menu item.

- **Enter Find String** and **Find Selection** menu items were added to the **Buffer** menu.

- The **Execution** menu has been renamed to the **Command** menu. The **Reset**, **Run**, **Step**, and **Clear** menu items have been removed. Ten user definable commands now displayed in this menu along with a **Set Commands...** menu item for defining them.