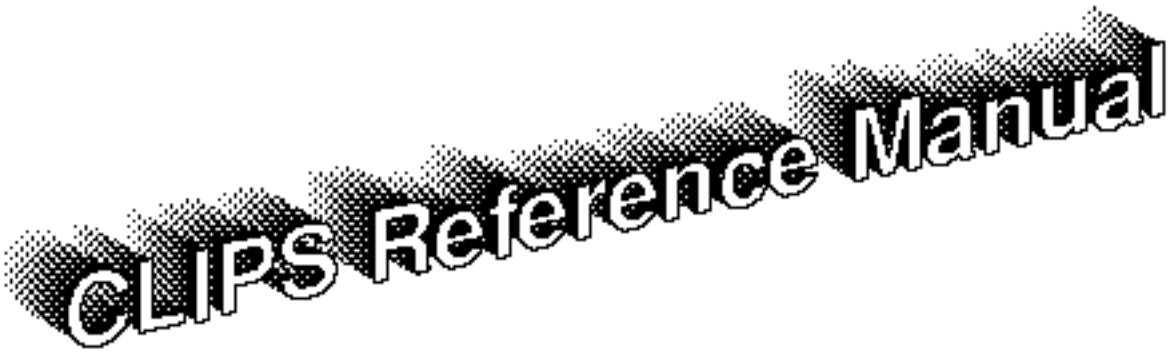
JSC-25012



Volume III
Interfaces Guide

CLIPS Version 6.0

June 2nd 1993

Software Technology Branch

Lyndon B. Johnson Space Center

CLIPS Interfaces Guide

Version 6.0 June 2nd 1993

CONTENTS

Preface vii

Acknowledgements xi

Section 1 - Introduction 13

Section 2 - CLIPS Windows Interface 15

2.1 The File Menu 16
2.1.1 Load Constructs .. (^L) 16
2.1.2 Load Batch 17
2.1.3 Load Binary Image 17
2.1.4 Turn Dribble On 17
2.1.5 Save Binary 17
2.1.6 Editor... (^E) 18
2.1.7 Quit CLIPS (^Q) 18
2.2 The Edit Menu 18
2.2.1 Paste (^V) 18
2.2.2 Complete... (^J) 18
2.3 The Execution Menu 20
2.3.1 Reset (^U) 20
2.3.2 Run (^R) 20
2.3.3 Step (^T) 20
2.3.4 Watch... (^W) 21
2.3.5 Options 22
2.3.6 Preferences 23
2.3.7 Clear CLIPS 24
2.4 The Browse Menu 24
2.4.1 Defrule Manager 25
2.4.2 Deffacts Manager 26
2.4.3 Deftemplate Manager 27
2.4.4 Deffunction Manager 28
2.4.5 Defglobals Manager 29
2.4.6 Defgeneric Manager 30
2.4.7 Defclass Manager 33
2.4.8 Definstances Manager 35
2.4.9 Agenda Manager 36
2.5 The Window Menu 37
2.5.1 Facts Window 37
2.5.2 Agenda Window 37

2.5.3 Instances Window....37
2.5.4 Globals Window....37
2.5.5 Focus Window....38
2.5.6 All Above....38
2.5.7 None....38
2.5.8 Clear Dialog Window....38
2.6 The HELP Menu....38
2.6.1 About CLIPS 6.0....39
2.6.2 CLIPS Help....39
2.6.3 Command Complete (^J)....39
2.7 The CLIPS Editor....40
2.7.1 The File Menu....40
2.7.1.1 New (^N)....40
2.7.1.2 Open... (^O)....40
2.7.1.3 Save (^S)....40
2.7.1.4 Save As....41
2.7.1.5 Revert....41
2.7.1.6 Printer Setup....41
2.7.1.7 Print... (^P)....41
2.7.1.8 Quit Editor (^Q)....41
2.7.2 The Edit Menu....42
2.7.2.1 Undo (^Z)....42
2.7.2.2 Cut (^X)....42
2.7.2.3 Copy (^C)....42
2.7.2.4 Paste (^V)....42
2.7.2.5 Clear....42
2.7.2.6 Balance (^B)....43
2.7.2.7 Complete... (^J)....43
2.7.2.8 Set Font....43
2.7.3 The Buffer Menu....44
2.7.3.1 Find (^F)....44
2.7.3.2 Replace (^R)....44
2.7.3.3 Load Selection ... (^K)....45
2.7.3.4 Batch Selection....45
2.7.3.5 Load Buffer....46
2.7.4 The HELP Menu....46
2.7.4.1 About the Editor....46
2.7.4.2 CLIPS Help....46
CLIPS Help....46
2.7.4.3 Balance (^B)....46
2.7.4.4 Command Complete (^J)....46
2.8 Creating the Windows 3.1 Executables....47
2.8.1 Building the CLIPS Interface Executable Using Borland C++ 3.1....47

2.8.2 Building the CLIPS Editor using Borland C++ 3.1....48
2.8.3 Building an EasyWin version of CLIPS for Windows....49
2.8.4 Building the CLIPS DOS Executable Using Borland C++ 3.1....50

Section 3 - CLIPS Macintosh Interface....51
3.1 The Menu....52
3.1.1 About CLIPS....52
3.2 The File Menu....52
3.2.1 New (⌘N)....52
3.2.2 Open... (⌘O)....53
3.2.3 Load... (⌘L)....53
3.2.4 Load Batch....53
3.2.5 Load Binary....53
3.2.6 Turn Dribble On....53
3.2.7 Close (⌘W)....54
3.2.8 Save (⌘S)....54
3.2.9 Save As....54
3.2.10 Save Binary....54
3.2.11 Revert....55
3.2.12 Page Setup....55
3.2.13 Print....55
3.2.14 Quit (⌘Q)....55
3.3 The Edit Menu....55
3.3.1 Undo (⌘Z)....56
3.3.2 Cut (⌘X)....56
3.3.3 Copy (⌘C)....56
3.3.4 Paste (⌘V)....56
3.3.5 Clear....56
3.3.6 Balance (⌘B)....56
3.3.7 Complete... (⌘J)....57
3.3.8 Set Font....58
3.3.9 Clear Window....58
3.3.10 Preferences....59
3.4 The Buffer Menu....61
3.4.1 Find... (⌘F)....61
3.4.2 Find Again (⌘G)....62
3.4.3 Replace (⌘H)....62
3.4.4 Replace and Find Again (⌘D)....62
3.4.5 Replace All....62
3.4.6 Load Selection (⌘K)....62
3.4.7 Batch Selection (⌘M)....63
3.4.8 Load Buffer....63
3.5 The Execution Menu....63

3.5.1 Reset (⌘E)....63
3.5.2 Run (⌘R)....63
3.5.3 Step (⌘T)....64
3.5.4 Watch....64
3.5.5 Options....65
3.5.6 Clear CLIPS....66
3.6 The Browse Menu....66
3.6.1 The Module Menu....67
3.6.2 Defrule Manager....67
3.6.3 Deffacts Manager....69
3.6.4 Deftemplate Manager....69
3.6.5 Deffunction Manager....70
3.6.6 Defglobal Manager....71
3.6.7 Defgeneric Manager....72
3.6.8 Defclass Manager....74
3.6.9 Definstances Manager....76
3.6.10 Agenda Manager....77
3.7 The Window Menu....78
3.7.1 Dialog Window....79
3.7.2 Facts Window....79
3.7.3 Agenda Window....79
3.7.4 Instances Window....79
3.7.5 Globals Window....79
3.7.6 Focus Window....79
3.7.7 All Above....80
3.7.8 Titled and Untitled Edit Windows....80
3.8 Creating The Macintosh Executables....80
3.8.1 Building the CLIPS Generic Executable Using THINK C....80
3.8.2 Building the CLIPS Interface Executable Using THINK C....81
3.8.3 Building the CLIPS Generic Executable Using MPW C....82
3.8.4 Building the CLIPS Interface Executable Using MPW C....82

Section 4 - CLIPS X Window Interface....85
4.1 The Menu....85
4.2 The File Menu....86
4.2.1 Edit... (^V)....86
4.2.2 Complete... (^C)....86
4.2.3 Load... (^L)....87
4.2.4 Load Batch....87
4.2.5 Load Binary....88
4.2.6 Dribble... (^N)....88
4.2.7 Save Binary....88
4.2.8 Quit (^Q)....88

4.3 The Execution Menu....89
4.3.1 Reset (^E)....89
4.3.2 Run (^R)....89
4.3.3 Step (^T)....89
4.3.4 Watch....89
4.3.5 Options....91
4.3.6 Clear CLIPS (^K)....92
4.3.7 Clear Window (^N)....92
4.4 The Browse Menu....92
4.4.1 Module Menu....93
4.4.2 Defrule Manager....94
4.4.3 Deffacts Manager....95
4.4.4 Deftemplate Manager....96
4.4.5 Deffunction Manager....97
4.4.6 Defglobal Manager....98
4.4.7 Defgeneric Manager....99
4.4.8 Defclass Manager....101
4.4.9 Definstances Manager....103
4.4.10 Agenda Manager....104
4.5 The Window Menu....105
4.5.1 Facts Window....105
4.5.2 Agenda Window....105
4.5.3 Instances Window....105
4.5.4 Globals Window....105
4.5.5 Focus Window....106
4.5.6 All....106
4.5.7 None....106
4.5.8 Command Line CLIPS (^Z)....106
4.5.9 Color Utility....106
4.5.9.1 How To Use The Color Utility....106
4.6 The Editor....108
4.6.1 File....108
4.6.1.1 Save....108
4.6.1.2 Save As....108
4.6.1.3 Revert....109
4.6.1.4 Load Selection....109
4.6.1.5 Batch Selection....109
4.6.1.6 Load Buffer....109
4.6.1.7 Exit....109
4.6.2 Edit....110
4.6.2.3 Complete... (^C)....110
4.6.2.2 Cut....111
4.6.2.3 Paste....111

4.6.2.4 Search/Replace... (^S/^R).....112
4.6.2.5 Balance (^B).....112
4.6.2.6 Beginning of File.....112
4.6.2.7 End of File.....112
4.6.3 Font.....112
4.6.3.1 5x8.....113
4.6.3.2 6x10.....113
4.6.3.3 6x13bold.....113
4.6.3.4 7x13bold.....113
4.6.3.5 8x13bold.....113
4.6.3.6 9x15bold.....113
4.6.3.7 10x20.....113
4.6.3.8 12x24.....113
4.6.4 Help.....114
4.6.4.1 Key Bindings.....114
4.7 Building The CLIPS X Window Interface.....114

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 of the prototype was to gain useful insight and knowledge about the construction of expert system tools and to lay the groundwork for the construction of a fully usable tool. The CLIPS prototype had numerous shortcomings, however, it demonstrated the feasibility of the project concept. After additional development, it became apparent that sufficient enhancements to the prototype would produce a low cost expert system tool that would be ideal for the purposes of training. Another year of development and internal use went into CLIPS improving its portability, performance, and functionality. A reference manual and user's guide were written during this time.

The first release of CLIPS to groups outside of NASA, version 3.0, occurred 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.

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 over 4,000 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 available at a nominal cost through COSMIC, the NASA software distribution center (for more on COSMIC, see appendix E of the *Basic Programming Guide*).

CLIPS Version 6.0

Version 6.0 of CLIPS contains five major enhancements. First, instances of user-defined classes in COOL can be pattern-matched on the left-hand side of rules. Second, CLIPS now contains considerable support for knowledge based systems software engineering. Support is now provided for building modular systems and many of the features previously available in CRSV are now directly supported in CLIPS (such as constraint consistency among uses of the same variable). Third,

deftemplates can now have more than one multifield slot. Fourth, it is now possible to nest other conditional elements within a *not* conditional element and two new conditional elements, *exists* and *forall* , are supported. Fifth, a Windows 3.1 CLIPS interface is now available for PC compatible computers. In addition, MS-DOS 286 and 386 versions of CLIPS are available which can use extended memory. For a detailed listing of differences between versions 5.1 and 6.0 of CLIPS, refer to appendix D of the *Basic Programming Guide*.

CLIPS Documentation

Three 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 and is intended for people with little or no expert system experience.
    - *Volume I - Rules*, which provides an introduction to rule-based programming using CLIPS.
    - *Volume II - Objects*, which provides an introduction to object-oriented programming using COOL.
- The *CLIPS Architecture Manual* which provides a detailed description of the CLIPS software architecture. This manual describes each module of CLIPS in terms of functionality and purpose. It is intended for people with extensive programming experience who are interested in modifying CLIPS or who want to gain a deeper understanding of how CLIPS works.

Acknowledgements

As with any large project, CLIPS is the result of the efforts of numerous people. The primary contributors have been: Robert Savely, previous branch chief of the STB and now chief scientist of advanced software technology at JSC, who conceived the project and provided overall direction and support; Chris Culbert, current branch chief of the STB, who managed the project, wrote the original *CLIPS Reference Manual*, and designed the original version of CRSV; Gary Riley, who designed and developed the rule-based portion of CLIPS, co-authored the *CLIPS Reference Manual* and *CLIPS Architecture Manual*, and developed the Macintosh interface for CLIPS; Brian Donnell, who designed and developed the CLIPS Object Oriented Language (COOL), co-authored the *CLIPS Reference Manual* and *CLIPS Architecture Manual*, and developed the previous MS-DOS interfaces for CLIPS; Bebe Ly, who was responsible for maintenance and enhancements to CRSV and is now responsible for developing the X Window interface for CLIPS; Chris Ortiz, who developed the Windows 3.1 interface for CLIPS; Dr. Joseph Giarratano of the University of Houston-Clear Lake, who wrote the *CLIPS User's Guide*; and Frank Lopez, who wrote the original prototype version of CLIPS.

Many other individuals contributed to the design, development, review, and general support of CLIPS, including: Jack Aldridge, Carla Armstrong, 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, Cynthia Rathjen, Eric Raymond, Reza Razavipour, Marsha Renals, Monica Rua, Tim Saito, 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

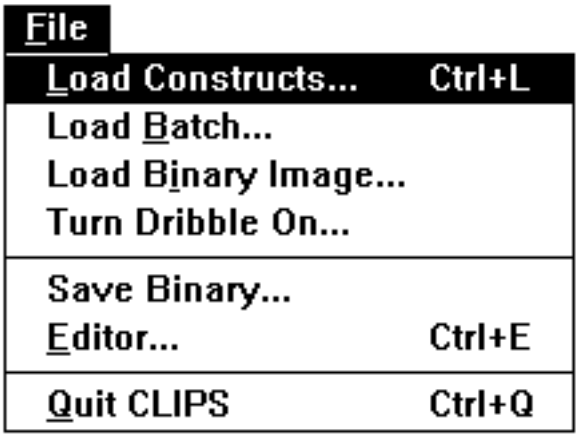
This section provides a brief summary of each menu command found in the CLIPS 6.0 Windows interface. The menus are listed in the left to right order in which they appear in the menu bar. The commands within each menu are also listed in the order in which they appear in the menu.

The rest of the CLIPS 6.0 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.

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 'C' key. If rules are being executed, then rule execution can be interrupted on a rule firing boundary by holding down the **control** key while pressing the 'C' 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 'C' 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 'C' key.

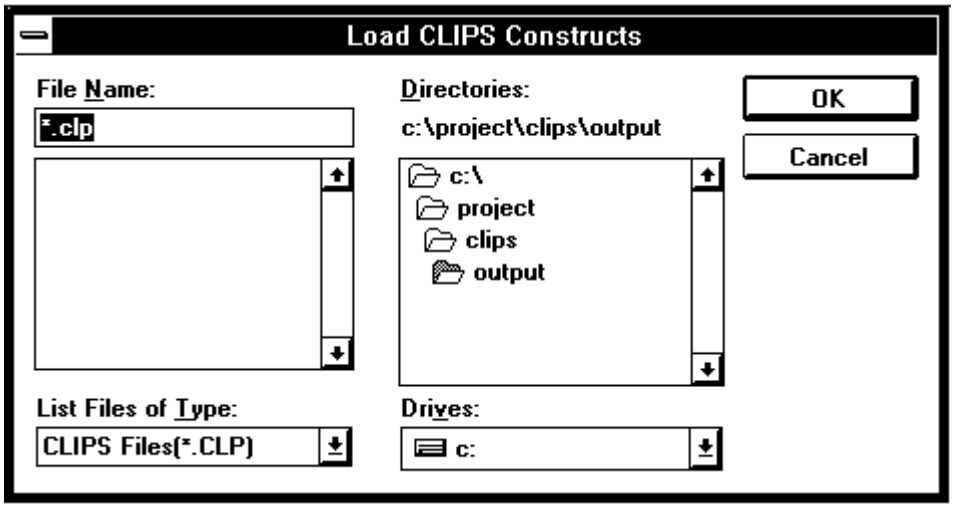
Some menu commands are available while CLIPS is executing a user program. Among these commands are the **Options...** menu item, the **Watch...** menu item, the **Turn Dribble On.../Turn Dribble Off** menu item, and all menu items in **Window** menu. The command summary for each menu item indicates whether 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



2.1.1 Load Constructs .. (^L)

This command displays the standard Windows 3.1 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")
```

2.1.2 Load Batch...

This command displays the standard 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.

2.1.3 Load Binary Image...

This command displays the standard Windows 3.1 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.

2.1.4 Turn Dribble On...

This command displays the standard Windows 3.1 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).

2.1.5 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.

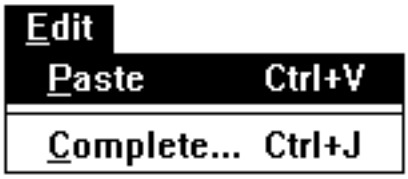
2.1.6 Editor... (^E)

This command will start an application used to edit CLIPS text files. This command is available when CLIPS is executing. For more information see Section 2.7 - The CLIPS Editor.

2.1.7 Quit CLIPS (^Q)

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

2.2 THE EDIT MENU



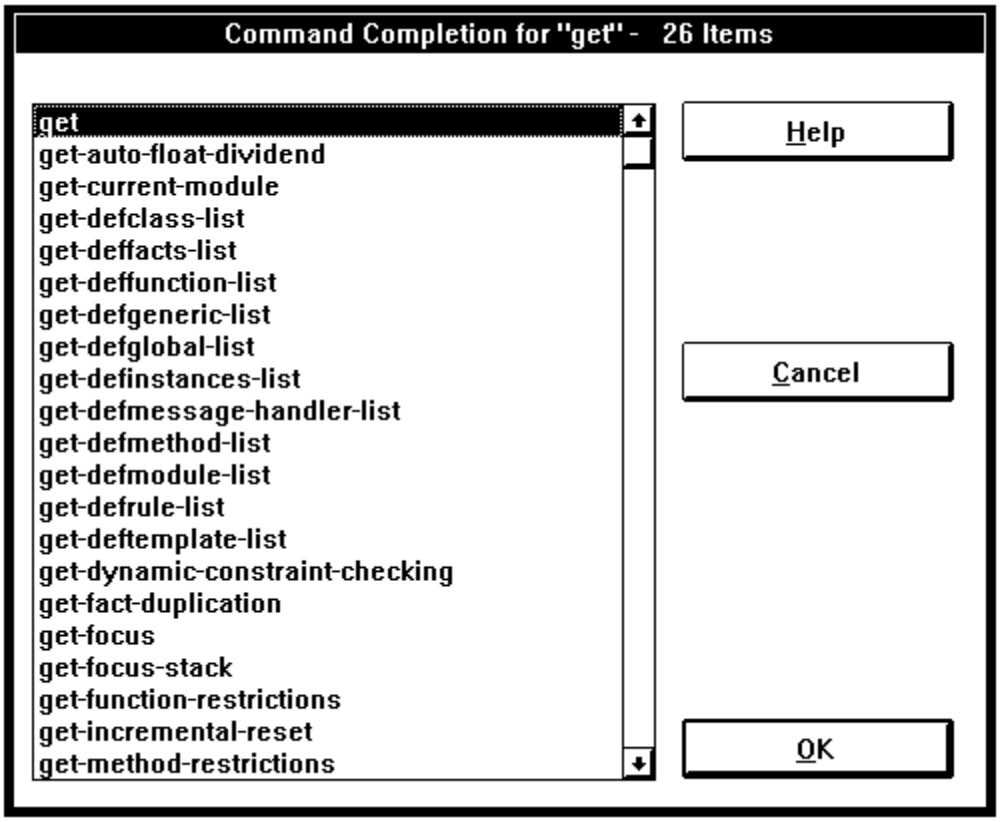
2.2.1 Paste (^V)

This command copies the contents of the Clipboard in the display window. Pasted text to the display window is always placed at the end of the display window regardless of whether the caret is. This command is not available when CLIPS is executing.

2.2.2 Complete... (^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).

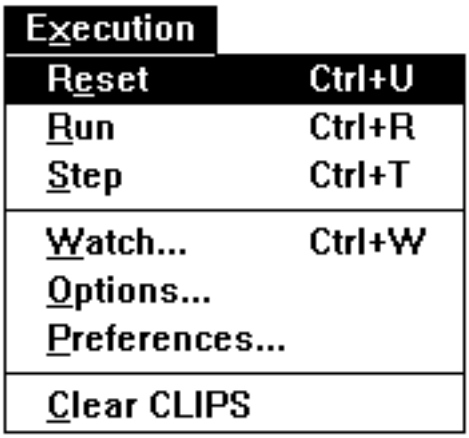
The following dialog shows the possible completion's for the symbol “get-”:



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.

2.3 THE EXECUTION MENU



2.3.1 Reset (^U)

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 Professor Joe 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.

2.3.2 Run (^R)

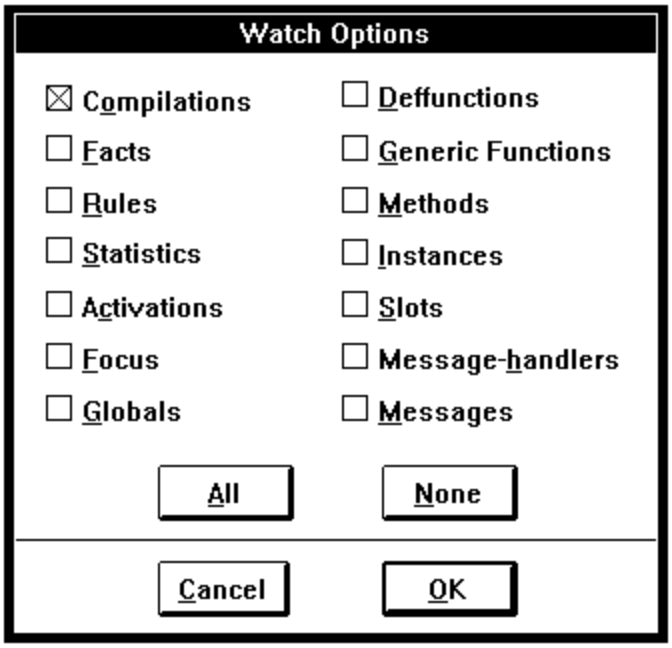
This command is equivalent to the CLIPS command (run). When this command is chosen, the CLIPS command (run) will be echoed to the dialog window and executed. During execution, this menu command is changed to **Halt** and can be selected to halt execution (as if a command period had been entered). Holding 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.

2.3.3 Step (^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.

2.3.4 Watch... (^W)

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

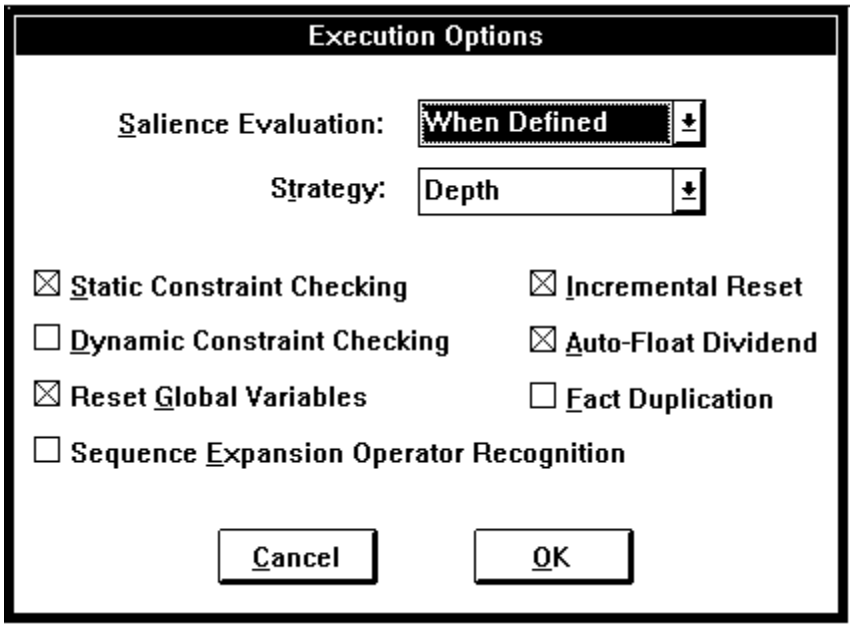


Watch items can be enabled or disabled by clicking the appropriate check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item check boxes. Pressing the **None** button unchecks all 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.

2.3.5 Options...

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

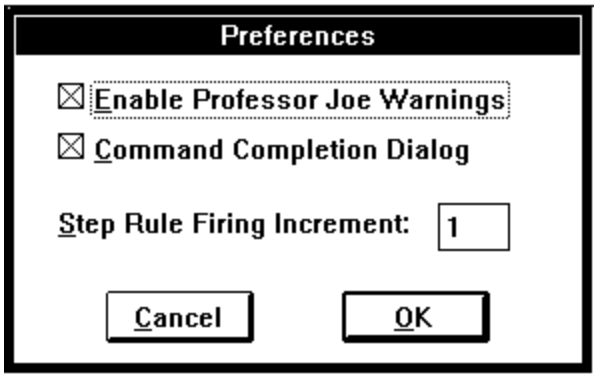


The **Salience Evaluation** pop-up menu allows the current salience evaluation behavior to be changed (to either when-defined, when-activated, or every-cycle). The **Strategy** pop-up menu allows the current conflict resolution strategy to be changed (to either depth, breadth, lex, mea, complexity, simplicity, 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.

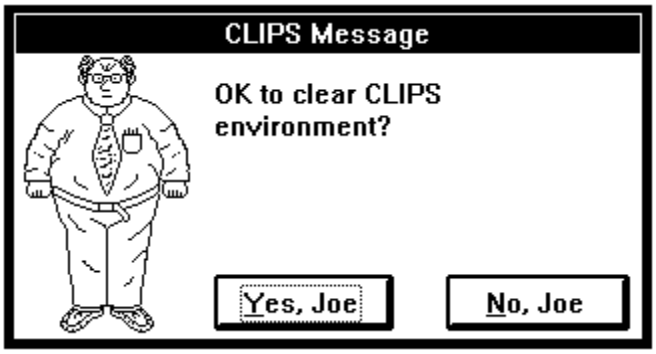
This command is available when CLIPS is executing.

2.3.6 Preferences...

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



Enable Professor Joe Warnings option determines whether certain warnings are issued. The default for this option is on. When Professor Joe 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 the Professor Joe feature on will display the following dialog box:



Professor Joe 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 Professor Joe Warnings are enabled. From the CLIPS command prompt, the command **set-joe-warnings** can be used to enable and disable Professor Joe Warnings (pass the symbol *TRUE* to enable warnings and the symbol *FALSE* to disable warnings).

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).

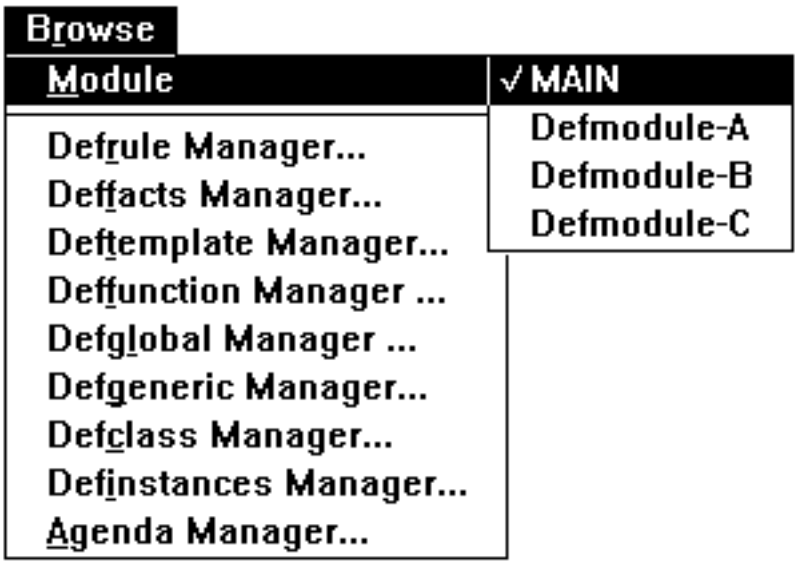
The **Step Rule Firing Increment** option determines how many rules will be executed when the **Step** command (see Execution menu) is issued. Any integer between 1 and 999 inclusive can be entered for this option. The default value for this option is 1.

This command is available when CLIPS is executing.

2.3.7 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 Professor Joe 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.

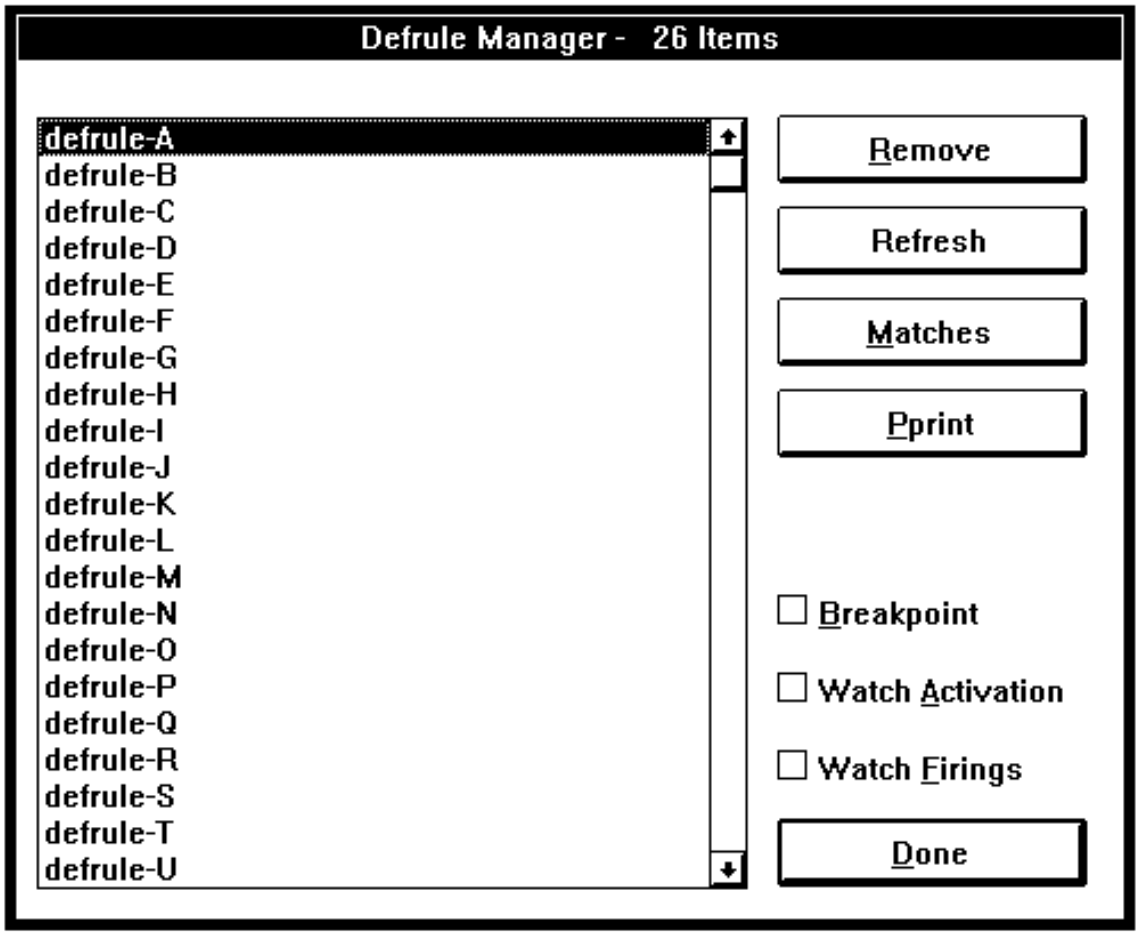
2.4 THE BROWSE MENU



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.

2.4.1 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.

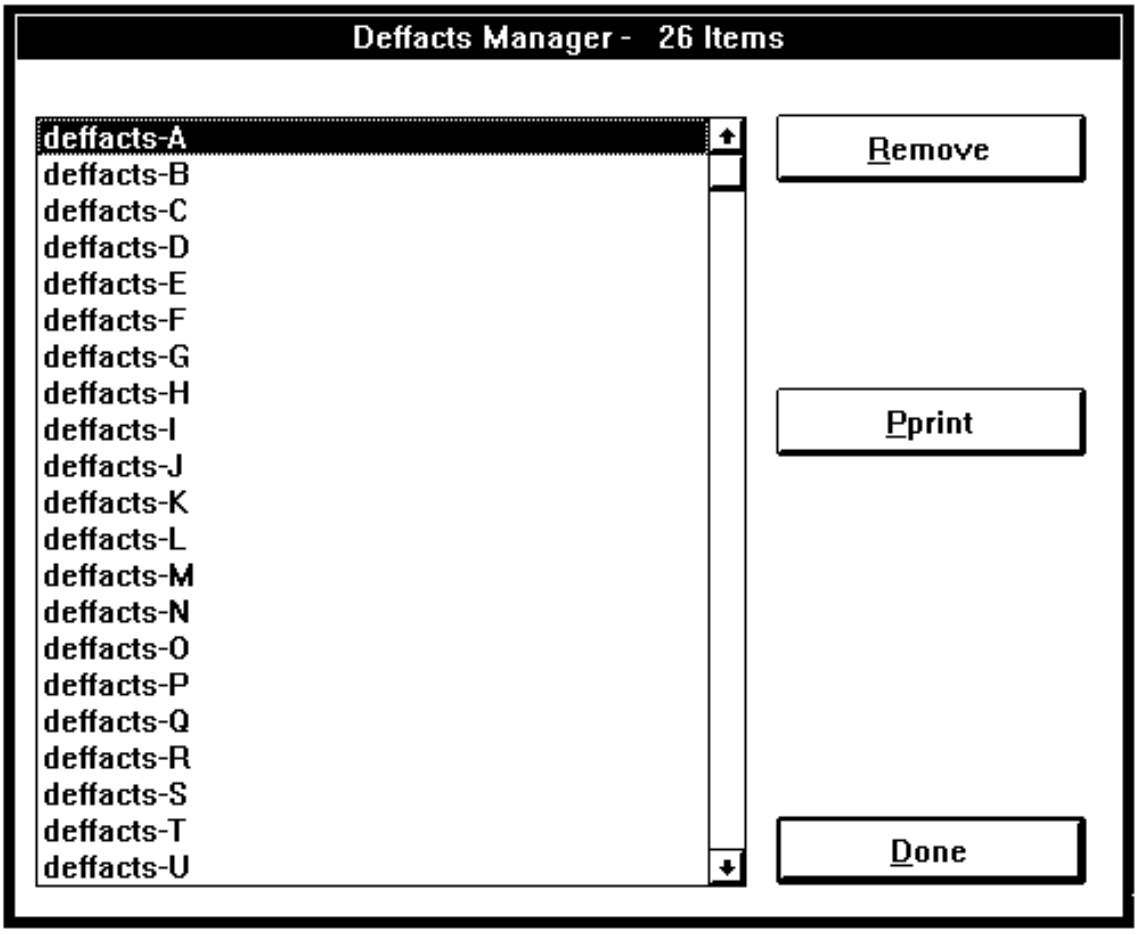
The **Breakpoint** check box is used to enable or disable a breakpoint on the currently selected defrule. Rules with a breakpoint set have a check in their check box while rules without a breakpoint have none. The **Watch Activations** check box is used to

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.

2.4.2 Deffacts Manager...

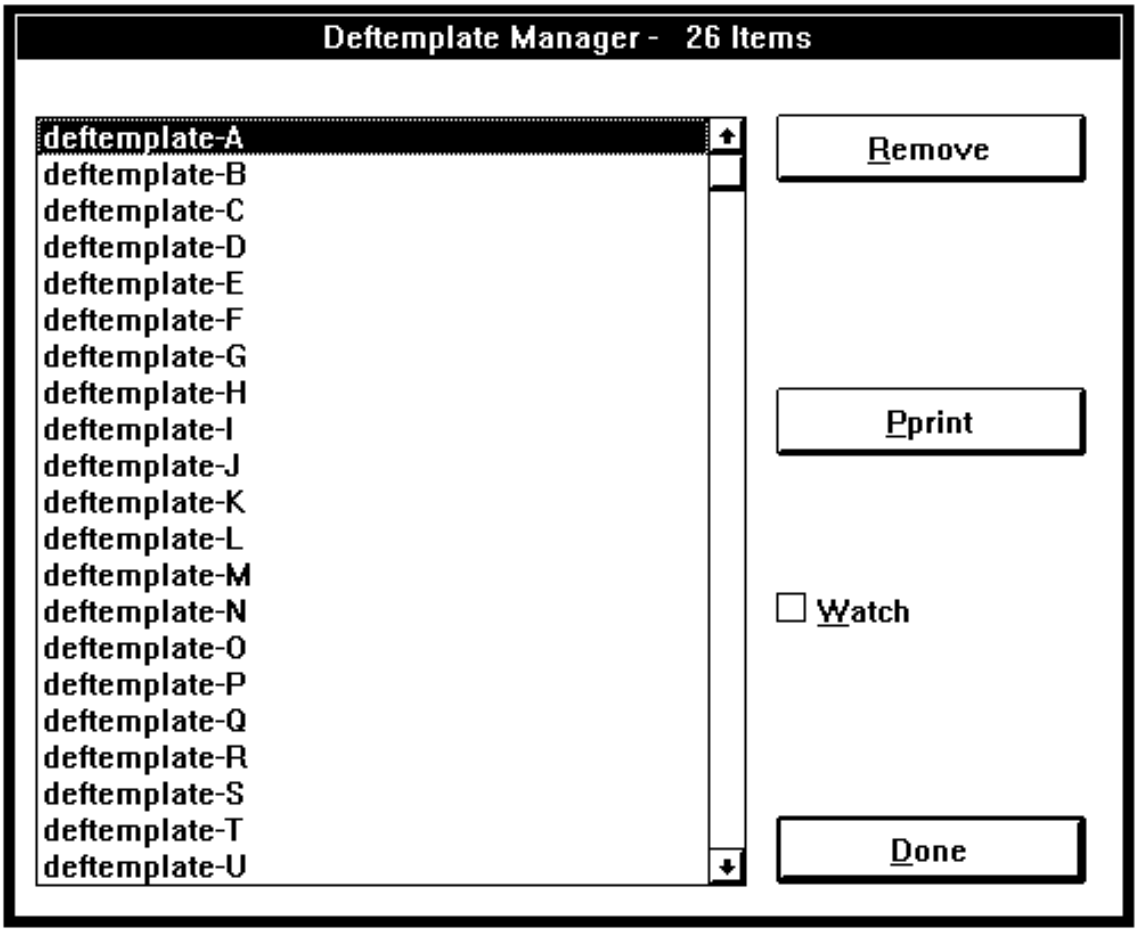
This command displays a dialog box which allows the deffacts in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffacts selected from the list of deffacts currently in the knowledge base.



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.

2.4.3 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

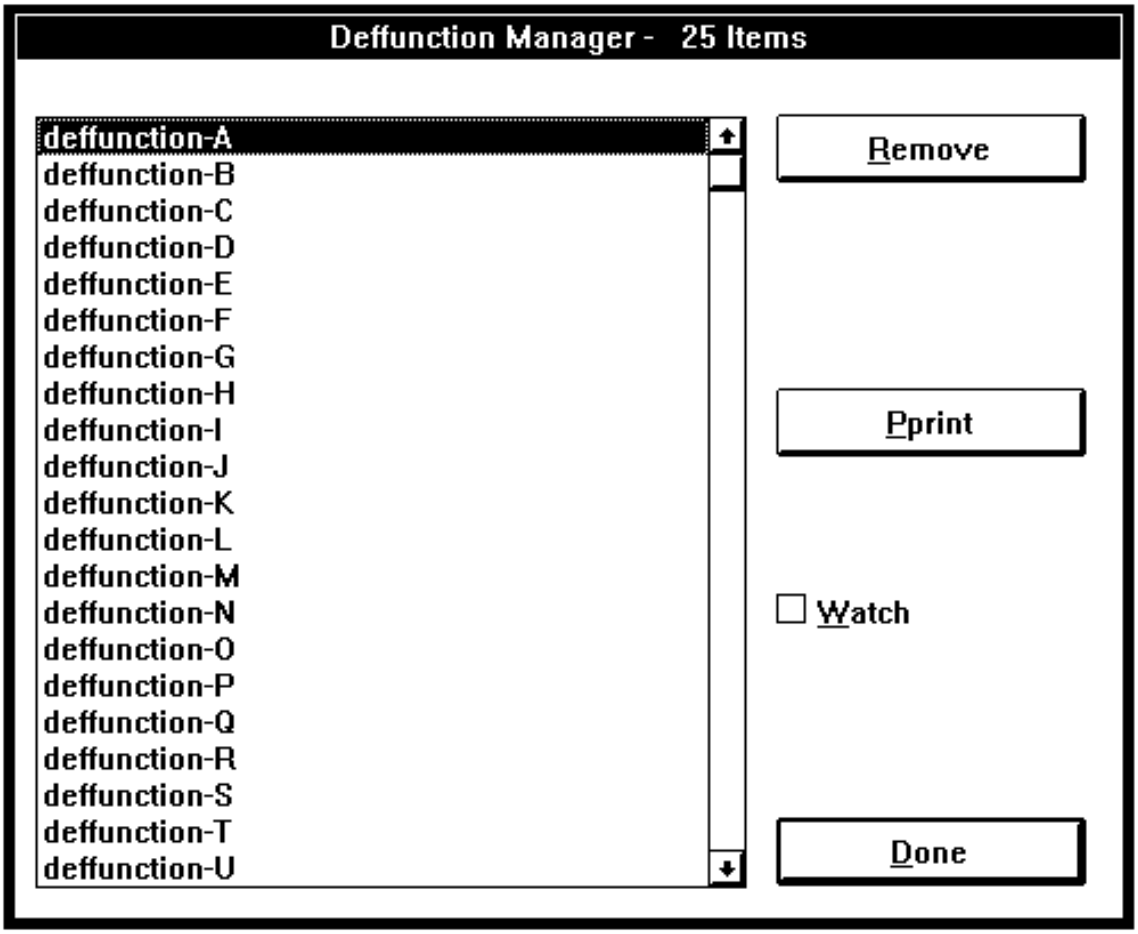
ppdeftemplate command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

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.

2.4.4 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.



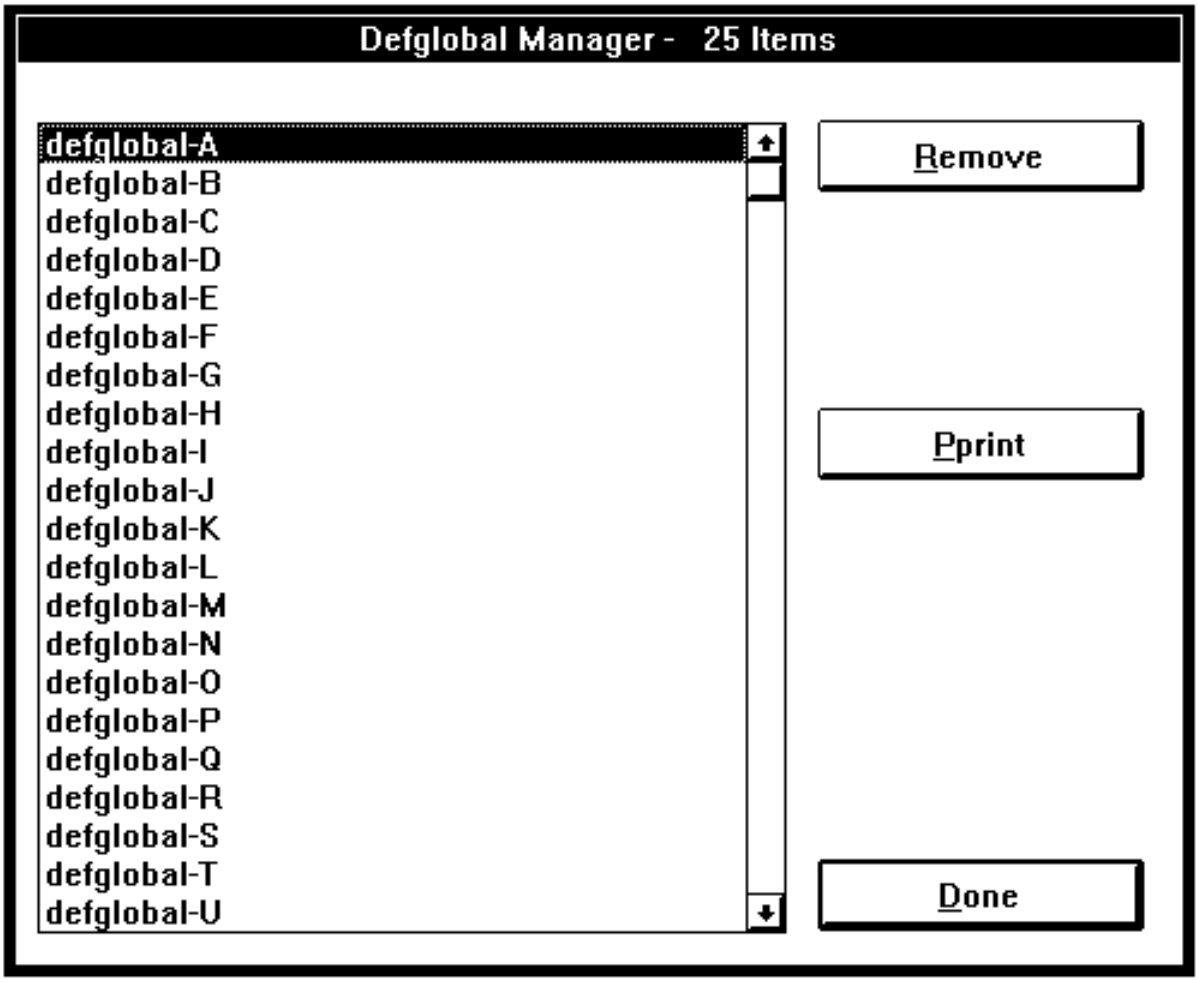
The **Remove** button will remove the currently selected deffunction. Pressing this button is equivalent to an **undeffunction** command. The **Pprint** button will pretty print the currently selected deffunction. Pressing this button is equivalent to a **ppdeffunction** command. The **Pprint** button 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.

2.4.5 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.



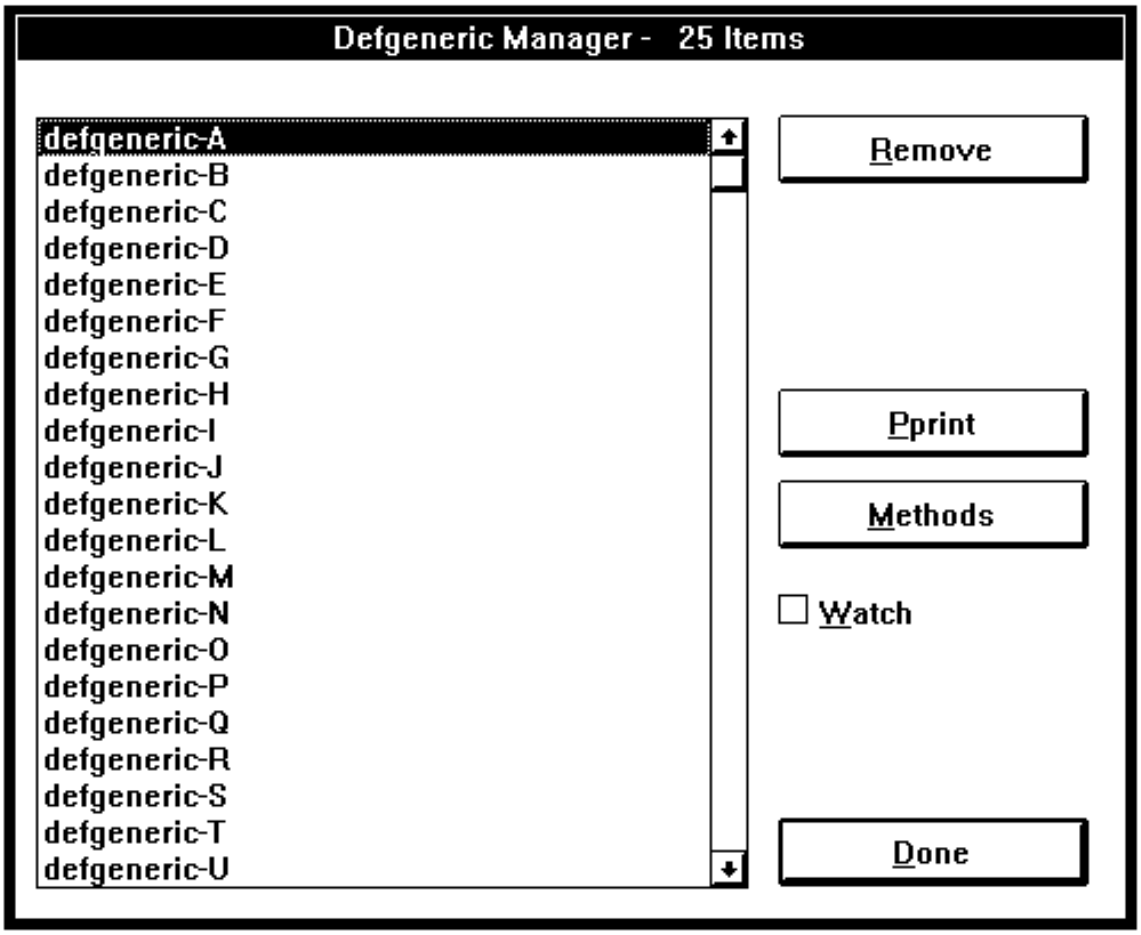
The **Remove** button will remove the currently selected defglobal. Pressing this button is equivalent to an **undefglobal** command. The **Pprint** button will pretty print the currently selected defglobal. Pressing this button is equivalent to a **ppdefglobal** command. The **Pprint** button echoes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

2.4.6 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

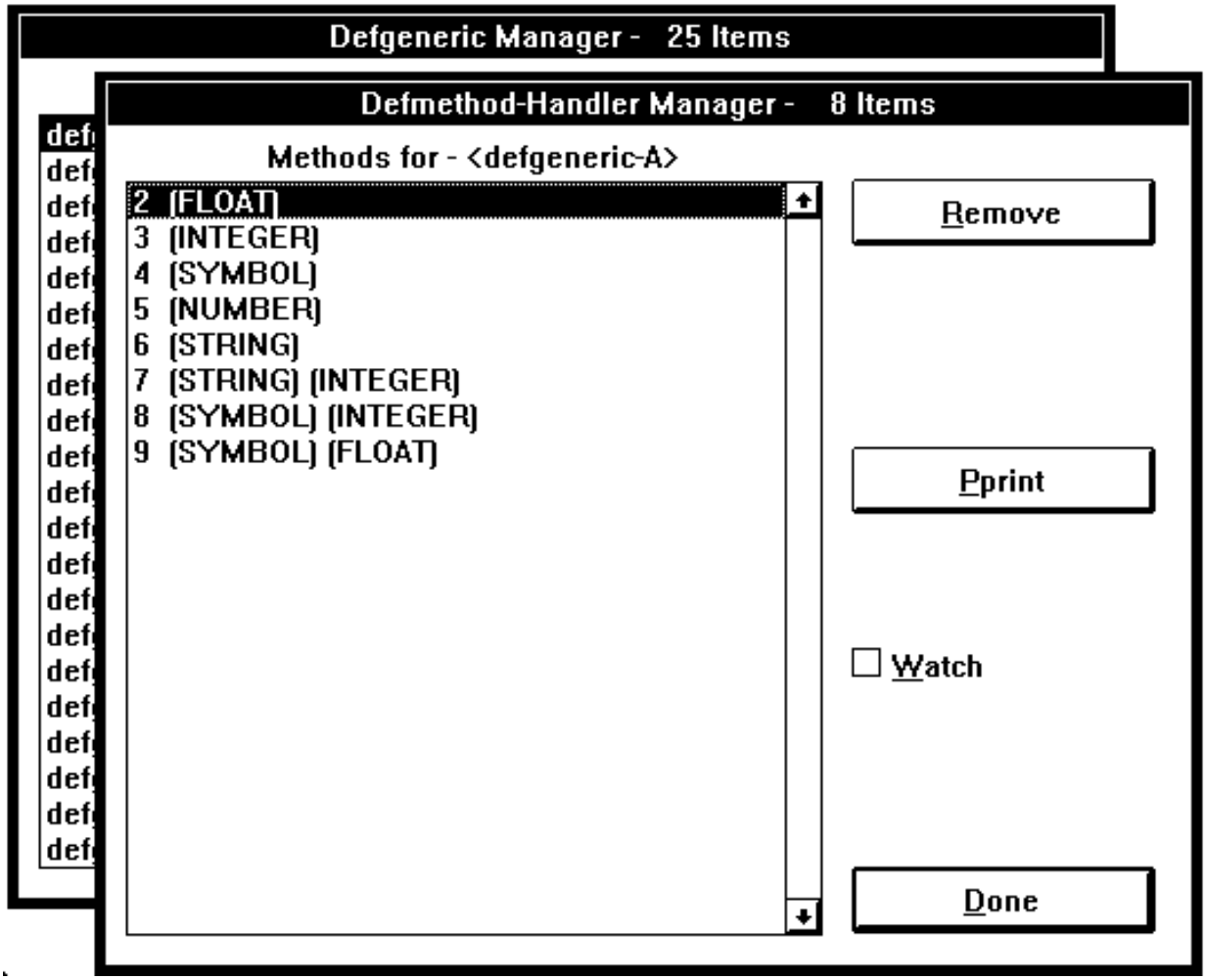
executed from this dialog. When finished, click the **Done** button to remove the dialog. The **Defgeneric Manager...** command is not available when CLIPS is executing.



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.



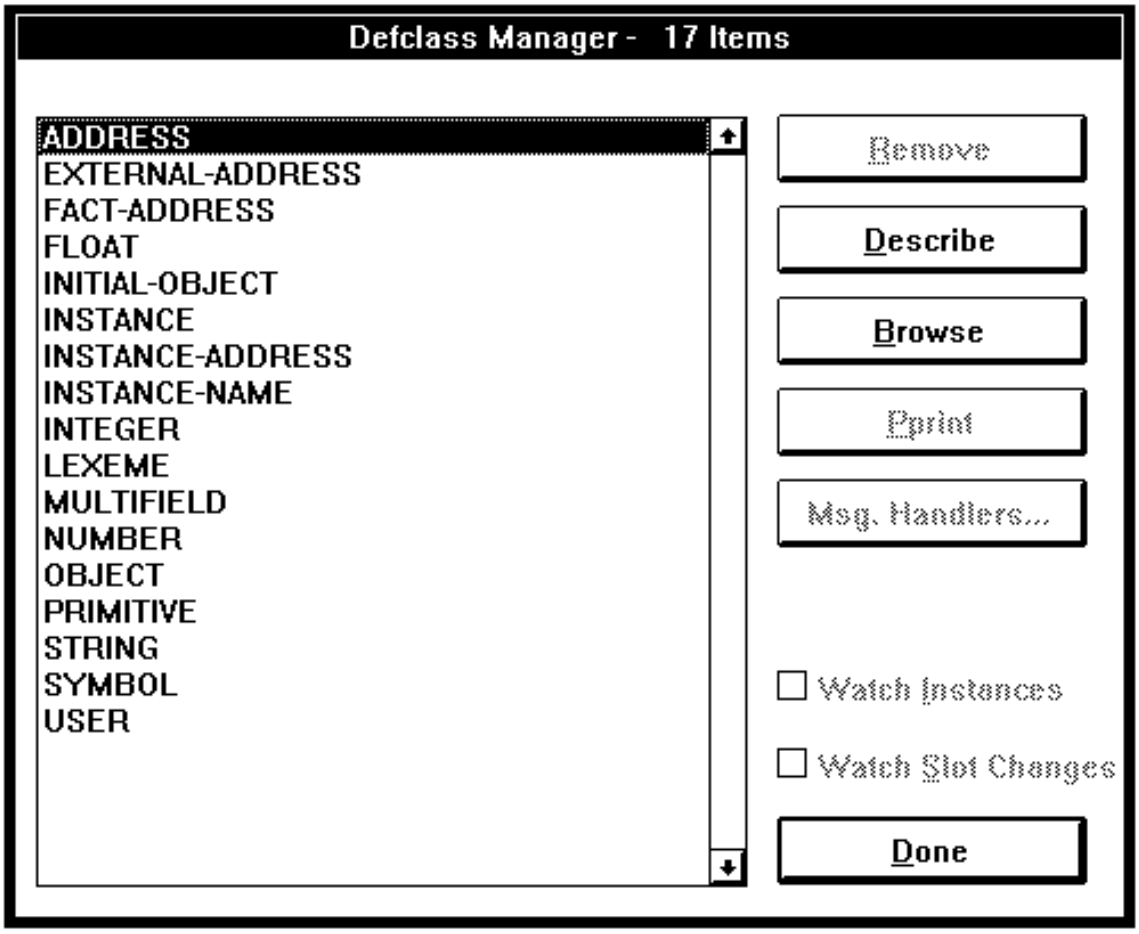
The **Remove** button will remove the currently selected defmethod. Pressing this button is equivalent to an **undefmethod** command. The **Pprint** button will pretty print the currently selected defmethod. Pressing this button is equivalent to a **ppdefmethod** command. The **Pprint** button echoes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

2.4.7 Defclass Manager...

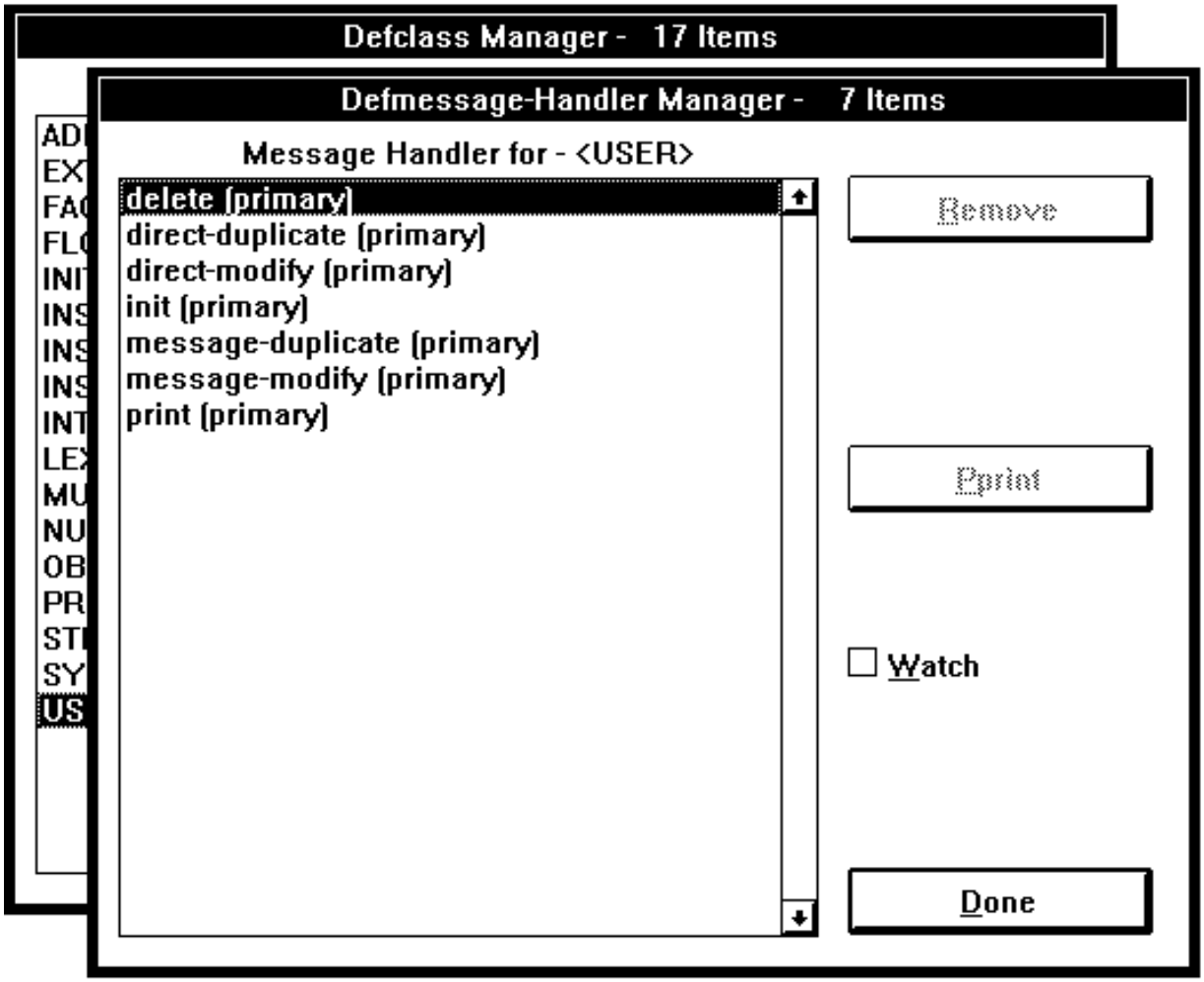
This command displays a dialog box which allows the defclasses in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defclass selected from the list of defclasses currently in the knowledge base. Any number of operations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

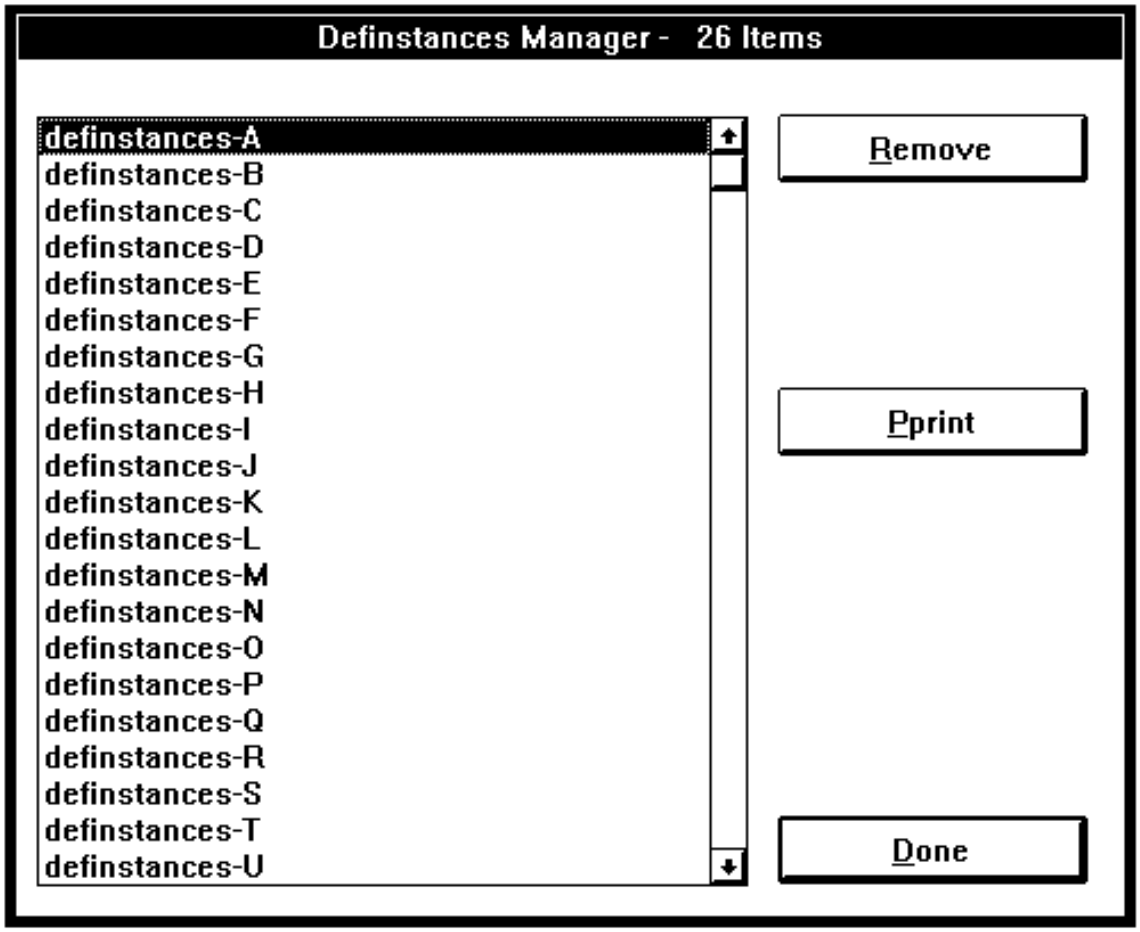
is equivalent to a **ppdefmessage-handler** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

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.

2.4.8 Definstances Manager...

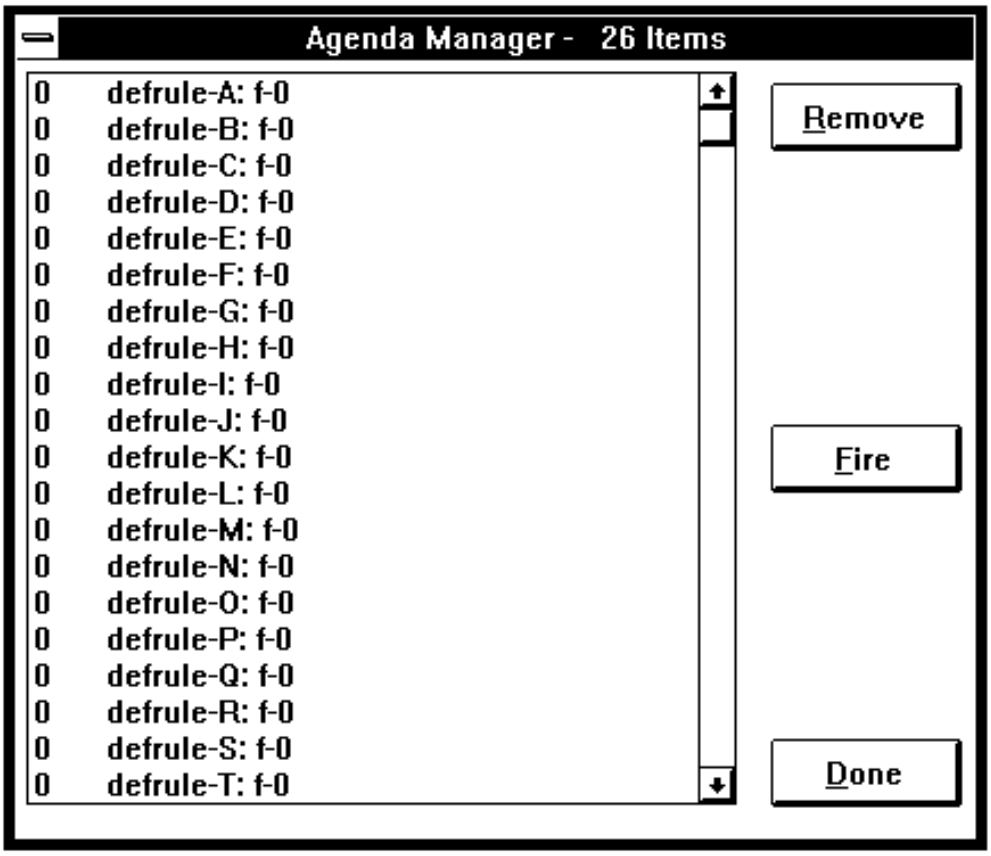
This command displays a dialog box which allows the definstances in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a definstances selected from the list of definstances currently in the knowledge base.



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.

2.4.9 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.



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.

2.5 THE WINDOW MENU

2.5.1 Facts Window

This command toggles the display of the **Facts Window.** The **Facts Window** displays the list and number of facts in the fact-list. A check mark is placed by the window name to indicate if it is visible. This command is available when CLIPS is executing.

2.5.2 Agenda Window

This command toggles the display of the **Agenda Window**. The **Agenda Window** displays the list of activation's currently on the agenda. A check mark is placed by the window name to indicated that it is visible. This command is available when CLIPS is executing.

2.5.3 Instances Window

This command toggles the display of 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 is visible. This command is available when CLIPS is executing.

2.5.4 Globals Window

This command toggles the display of 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 is visible. This command is available when CLIPS is executing.

2.5.5 Focus Window

This command toggles the display of the **Focus Window** The **Focus Window** displays the current focus stack. A check mark is placed by the window name to indicated that it is visible. This command is available when CLIPS is executing.

2.5.6 All Above

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

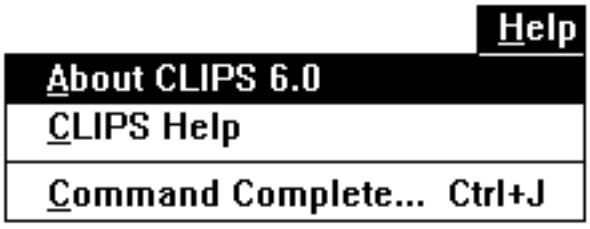
2.5.7 None

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

2.5.8 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 THE HELP MENU

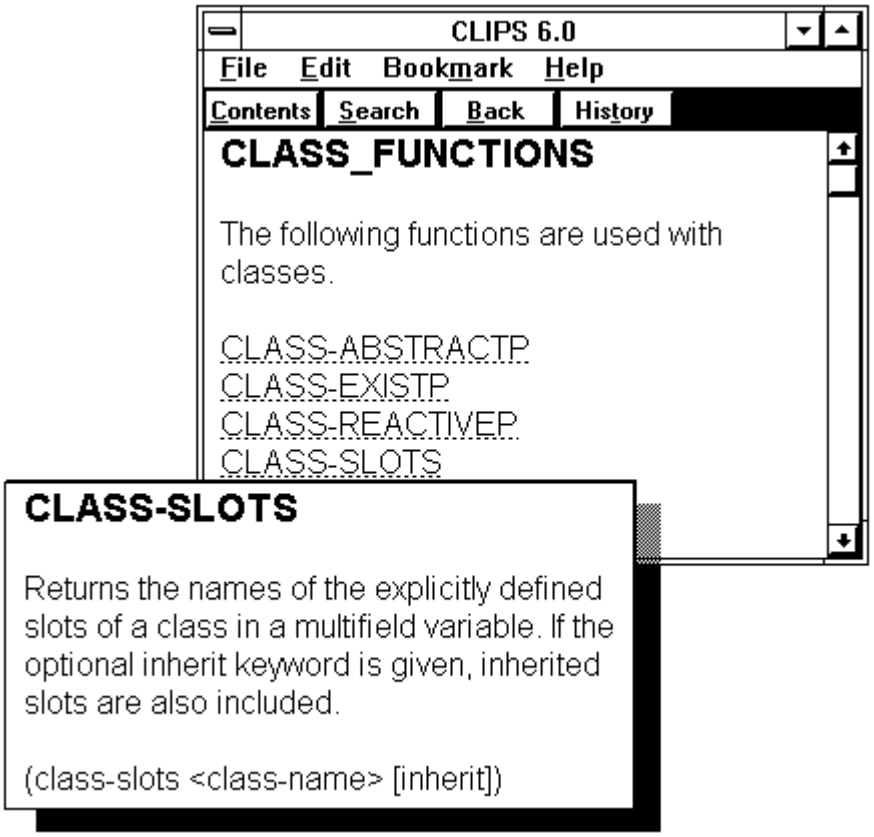


2.6.1 About CLIPS 6.0

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

2.6.2 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.

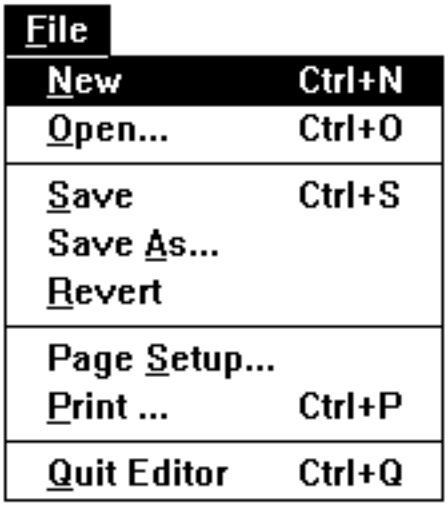


2.6.3 Command Complete (^J)

(See section 2.2.2). This command is not available when CLIPS is executing.

2.7 THE CLIPS EDITOR

2.7.1 The File Menu



2.7.1.1 New (^N)

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

2.7.1.2 Open... (^O)

This command displays the standard file selection dialog box, allowing the user to select a text file to be opened as a buffer for editing. Only one file can be opened at a time.

2.7.1.3 Save (^S)

This command saves the file in the active edit window. If the file is untitled, 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.

2.7.1.4 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.

2.7.1.5 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.

2.7.1.6 Printer Setup

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

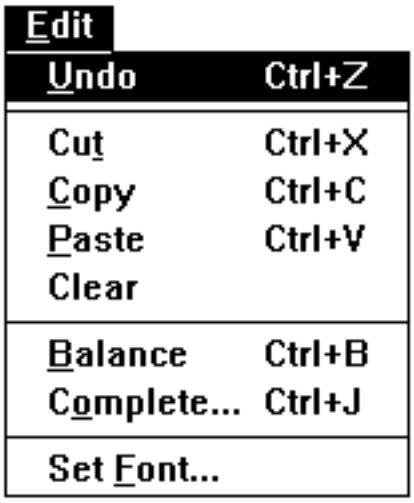
2.7.1.7 Print... (^P)

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

2.7.1.8 Quit Editor (^Q)

This command closes the active window if it has a go-away box in its upper left corner. If the active window is an editing buffer that has been modified since it was last saved, a dialog box will confirm whether the changes should be saved or discarded or whether the close command should be canceled.

2.7.2 The Edit Menu



2.7.2.1 Undo (^Z)

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

2.7.2.2 Cut (^X)

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

2.7.2.3 Copy (^C)

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

2.7.2.4 Paste (^V)

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

2.7.2.5 Clear

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

2.7.2.6 Balance (^B)

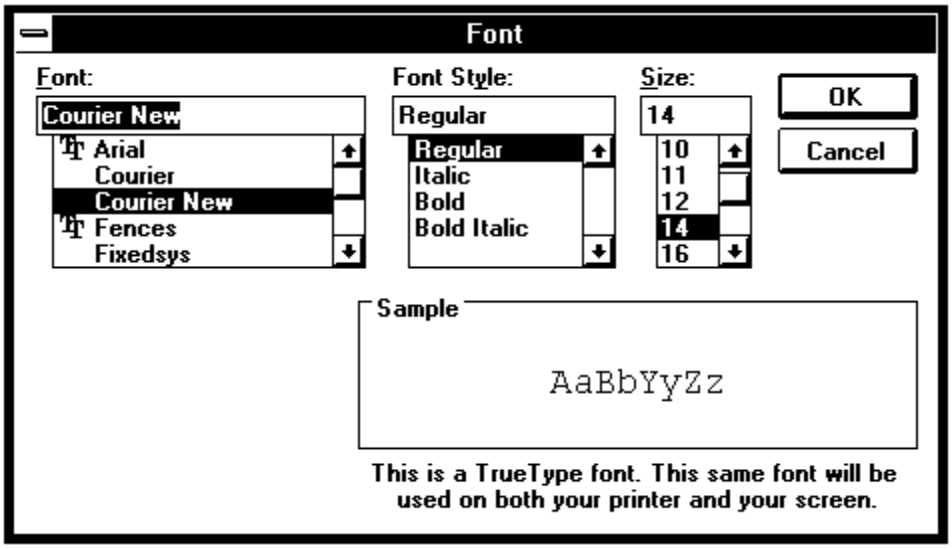
This command operates on the current selection in the edit window by attempting to find the smallest selection containing the current selection which has balanced parentheses. Repeatedly using this command will select larger and larger selections of text until a balanced selection cannot be found. The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

2.7.2.7 Complete... (^J)

(See section 2.2.2). This command is not available when CLIPS is executing.

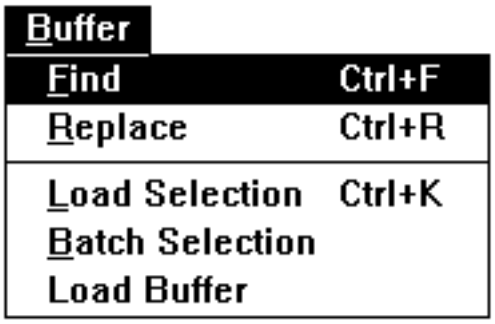
2.7.2.8 Set Font...

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:



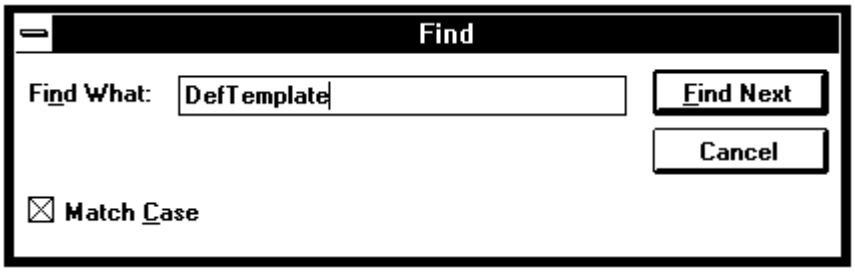
Select a font and font size from the popup menus on the left of the dialog. The sentence at the bottom of the dialog will be displayed using the selected font. Click the **OK** button to change the font used in the currently active editing buffer. Click the **Cancel** button to retain the current font setting.

2.7.3 The Buffer Menu



2.7.3.1 Find (^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.

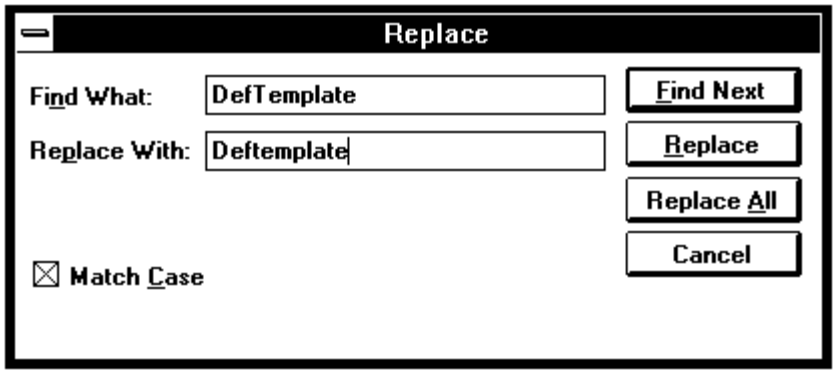


The **Match Case** option makes the string search operation case-insensitive for alphabetic characters; that is, the string “Upper” will not match the string “uPPER”. The default for this setting is off.

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.

2.7.3.2 Replace (^R)

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.

2.7.3.3 Load Selection ... (^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.

2.7.3.4 Batch Selection

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.

2.7.3.5 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.

2.7.4 THE HELP MENU

2.7.4.1 About the Editor

This command displays version information about the CLIPS Editor. Click the OK button to continue.

2.7.4.2 CLIPS Help

CLIPS Help

(See section 2.6.2.).

2.7.4.3 Balance (^B)

(See section 2.7.2.6)

2.7.4.4 Command Complete (^J)

(See section 2.2.2). This command is not available when CLIPS is executing.

2.8 CREATING THE WINDOWS 3.1 EXECUTABLES

2.8.1 Building the CLIPS Interface Executable Using Borland C++ 3.1

The following steps describe how to create the Windows 3.1 machine specific version of CLIPS using Borland C++ 3.1:

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

clipsdde.h	diag_win.h	dialog.h	ids.h
main.h	menucmds.h	status.h	
clipsdde.c	diag_win.c	dialog1.c	dialog2.c
main.c	menu.c	menucmds.c	status.c
system.c			
clipswin.def	clipswin.rc	clipswin.ico	joe.bmp
wait.cur	query.cur	1.cur	2.cur
3.cur	4.cur	5.cur	6.cur
7.cur	8.cur	9.cur	

2) From DOS, Create a directory named **C:\CLIPS**.

3) Create a sub-directory named **SOURCE** in the **CLIPS** directory. Copy all of the CLIPS source code into the **C:\CLIPS\SOURCE** directory.

4) Create a sub-directory named **INTRFACE** in the **CLIPS** directory. Windows interface source code (the .h and .c source listed in step 1) into the **C:\CLIPS\INTRFACE** directory.

5) Edit the **setup.h** file. Set the compiler directive flags IBM_TBC to 1, WINDOW_INTERFACE to 1, CLP_EDIT to 0, and BLOCK_MEMORY to 1.

6) Create a project (CLIPSWIN.PRJ provided) using the Borland C++ for Windows project editor (BCW.EXE). Add all the appropriate files (*.C, *.H, *.RC, *.DEF). Note: Both source and Interface directories contain the file **main.c**. Use only the **main.c** from the interface directory.

7) Select **Directories** in the **Options** Menu. Create a path to the include directories **C:\BORLANDC\INCLUDE;C:\CLIPS\SOURCE;C:\CLIPS\INTRFACE**.

Create a path to the library directory **C:\BORLANDC\LIB**. Create a path to the output directory **C:\CLIPS**.

8) Set the following compiler options. For **Code Generation**: Large Memory Model, Never Assume SS Equals DS, Treat enums as ints, Duplicate strings merged, and Pre-Compiled headers. For **Entry and Exit Code Generation**, set Windows explicit functions exported. For **Advanced Code Generation**: Automatic Far Data (Checked) and Far Data Threshold set to 10 Bytes. Use the default **Optimization Options**.

9) Select the **Output Windows EXE** linker option.

10) Under the **Compile** Menu Choose **Build All**.

2.8.2 Building the CLIPS Editor using Borland C++ 3.1

The following steps describe how to create the Windows machine specific version of CLIPS using Borland C++ 3.1.

1) The following source files are needed to build the clips editor.

ide_bal.h	ide_dde.h	ide_ids.h	ide_misc.h
ide_srch.h			
ide_bal.c	ide_dde.c	ide_main.c	ide_misc.c
ide_srch.c			
ide_clp.ico	ide_clps.def	ide_clps.rc	

2) From DOS, Create a directory named **C:\CLIPS**.

3) Create a sub-directory named **EDITOR** in the **CLIPS** directory. Copy all of the Windows editor source code (the .h and .c source listed in step 2) into the **C:\CLIPS\EDITOR** directory.

4) Create a project (IDE_CLPS.PRJ provided) using the Borland C++ for Windows project editor (BCW.EXE). Add all the appropriate files (*.C, *.H, *.RC, *.DEF).

5) Select **Directories** in the **Options** Menu. Create a path to the include directories **C:\BORLANDC\INCLUDE;C:\CLIPS\EDITOR**. Create a path to the library directory **C:\BORLANDC\LIB**. Create a path to the output directory **C:\CLIPS**.

6) Set the following compiler options. For **Code Generation**: Large Memory Model, Assume SS Equals DS (default for memory model), and Duplicate strings merged. For **Entry and Exit Code Generation**, set Windows smart callbacks. Use the default **Optimization Options**.

7) Select the **Output Windows EXE** linker option.

8) Choose **Compile** and **Build All**.

2.8.3 Building an EasyWin version of CLIPS for Windows.

The following steps describe how to create the Windows 3.1 machine specific version of CLIPS using Borland C++ 3.1 and EasyWin

1) From DOS, Create a directory named **C:\CLIPS**.

2) Create a sub-directory named **SOURCE** in the **CLIPS** directory. Copy all of the CLIPS source code into the **C:\CLIPS\SOURCE** directory. Copy the **clipswin.def** and the **system.c** file from the INTRFACE directory.

3) Edit the **setup.h** file. Set the compiler directive flags IBM_TBC to 1, WINDOW_INTERFACE to 0, CLP_EDIT to 0, and BLOCK_MEMORY to 1.

4) Create a project (CLIPSCMD.PRJ provided) using the Borland C++ for Windows project editor (BCW.EXE). Add all the appropriate files (*.C, *.H, *.RC, *.DEF). Note: Both source and Interface directories contain the file **main.c**. Use only the **main.c** from the SOURCE directory.

6) Select **Directories** in the **Options** Menu. Create a path to the include directories **C:\BORLANDC\INCLUDE;C:\CLIPS\SOURCE**. Create a path to the library directory **C:\BORLANDC\LIB**. Create a path to the output directory **C:\CLIPS**.

7) Set the following compiler options. For **Code Generation**: Large Memory Model, Never Assume SS Equals DS, Treat enums as ints, Duplicate strings merged, and Pre-Compiled headers. For **Entry and Exit Code Generation**, set Windows explicit functions exported. For **Advanced Code Generation**: Automatic Far Data (Checked) and Far Data Threshold set to 10 Bytes. Use the default **Optimization Options**.

8) Select the **Output Windows EXE** linker option.

9) Under the **Compile** Menu Choose **Build All**.

2.8.4 Building the CLIPS DOS Executable Using Borland C++ 3.1

The following steps describe how to create the command line executable version of CLIPS using Borland C++ 3.1 for DOS and EasyWin.

1) From DOS, Create a directory named **C:\CLIIPS**.

2) Create a sub-directory named **SOURCE** in the **CLIPS** directory. Copy all of the CLIPS source code into the **C:\CLIPS\SOURCE** directory (including **main.c**).

3) Edit the **setup.h** file. Set the compiler directive flags IBM_TBC to 1, WINDOW_INTERFACE to 0, CLP_EDIT to 0, and BLOCK_MEMORY to 1.

4) Create a project (CLIPS.PRJ provided) using the Borland C++ for Windows project editor (BC.EXE). Add all the appropriate files (*.C and *.H).

5) Select **Directories** in the **Options** Menu. Create a path to the include directory **CC:\BORLANDC\INCLUDE;C:\CLIPS\SOURCE**. Create a path to the library directory **C:\BORLANDC\LIB**. Create a path to the output directory **C:\CLIPS**.

6) Set the following compiler options. For **Code Generation**: Large Memory Model, Never Assume SS Equals DS, Treat enums as ints, Duplicate strings merged, and Pre-Compiled headers. For **Entry and Exit Code Generation**, select DOS Standard. For **Advanced Code Generation**, check Automatic Far Data and set the Far Data Threshold to 10 bytes. Use the default **Optimization Options**.

7) Select the **Standard DOS EXE** linker option.

8) Choose **Compile** and **Build All**.

Section 3 - CLIPS Macintosh Interface

This section provides a brief summary of each menu command found in the CLIPS 6.0 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 rest of the CLIPS 6.0 Macintosh interface is straightforward, following Macintosh 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 Macintosh editors.

Upon startup, CLIPS will look for a text or batch file named “CLIPS Prefs” first in the folder in which CLIPS is contained and second in the system folder. If such a file is found, it will automatically be executed as a batch file. This provides a convenient method for executing a series of command automatically when CLIPS is started.

If rules are not being executed, then execution of the current command or CLIPS program can be halted by holding down the ⌘ key while pressing the period key. If rules are being executed, then rule execution can be interrupted on a rule firing boundary by holding down the ⌘ key while pressing the period key. Holding down the shift key, the ⌘ 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 ⌘ 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 ⌘ 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 resized while CLIPS is executing.

3.1 THE MENU

3.1.1 About CLIPS...

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

3.2 THE FILE MENU

3.2.1 New (⌘N)

This command opens a new buffer for editing with the window name untitled. This command is not available when CLIPS is executing.

3.2.2 Open... (⌘O)

This command displays the standard file selection dialog box, 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. This command is not available when CLIPS is executing.

3.2.3 Load... (⌘L)

This command displays the standard 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.

3.2.4 Load Batch...

This command displays the standard 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.

3.2.5 Load Binary...

This command displays the standard 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.

3.2.6 Turn Dribble On...

This command displays the standard 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).

3.2.7 Close (⌘W)

This command closes the active window if it has a go-away box in its upper left corner. If the active window is an editing buffer that has been modified since it was last saved, a dialog box will confirm whether the changes should be saved or discarded or whether the close command should be canceled. This command is available when CLIPS is executing. This command is available when CLIPS is executing.

3.2.8 Save (⌘S)

This command saves the file in the active edit window. If the file is untitled, a dialog box will prompt for a file name under which to save the file. The **Batch File** 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 a CLIPS editing buffer. This command is available when CLIPS is executing.

3.2.9 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. This command is available when CLIPS is executing.

3.2.10 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.

3.2.11 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. This command is available when CLIPS is executing.

3.2.12 Page Setup...

This command allows the user to specify information about the size of paper used by the printer. This command is available when CLIPS is executing.

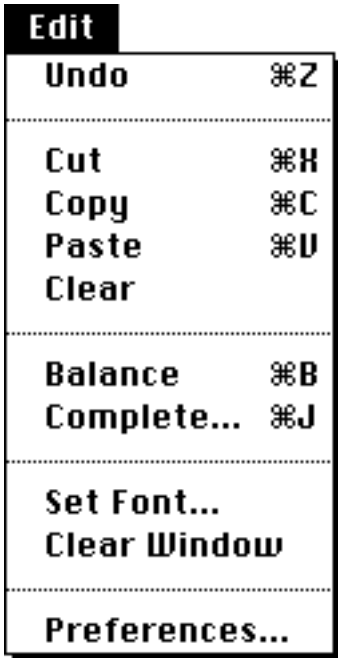
3.2.13 Print...

This command allows the user to print the active edit window. This command is available when CLIPS is executing.

3.2.14 Quit (⌘Q)

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

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 clear 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**. If the **Paste** command is undone, the menu item will read **Redo Paste**. This command is not available when CLIPS is executing.

3.3.2 Cut (⌘X)

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.

3.3.3 Copy (⌘C)

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

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

3.3.5 Clear

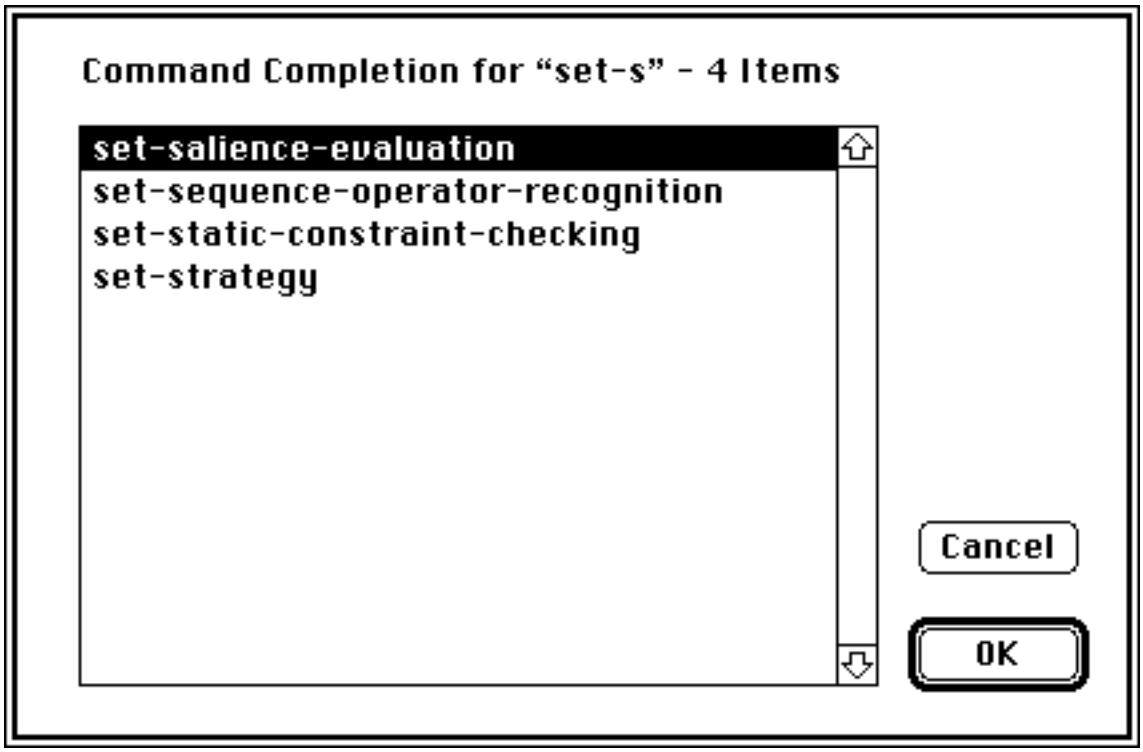
This command removes selected text in the active edit window. The selected text is not placed in the Clipboard. This command is not available when CLIPS is executing.

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

3.3.7 Complete... (⌘J)

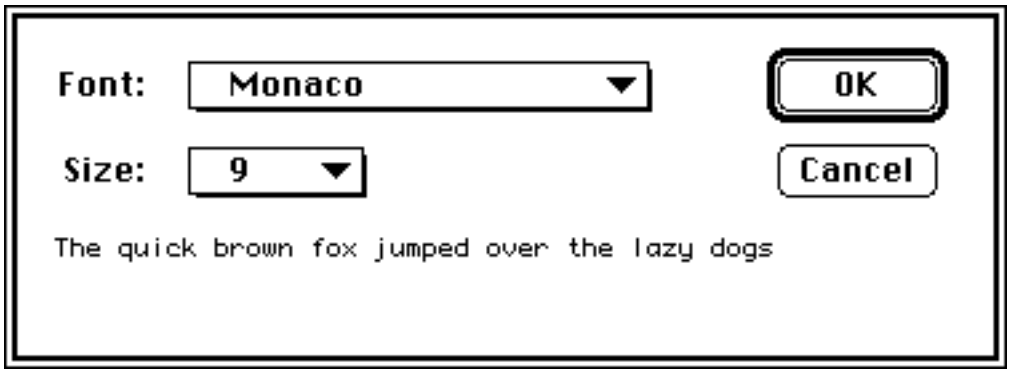
This command “completes” the symbol currently being entered in the display window or the active edit window. In the display window, the symbol to be completed is always the last symbol in the display window (the current selection has no effect). In an edit window, the symbol being completed is the current 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 (or optionally a beep is sounded—See section 3.3.10). The following dialog shows the possible completions for the symbol “set-s”:



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.

3.3.8 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:



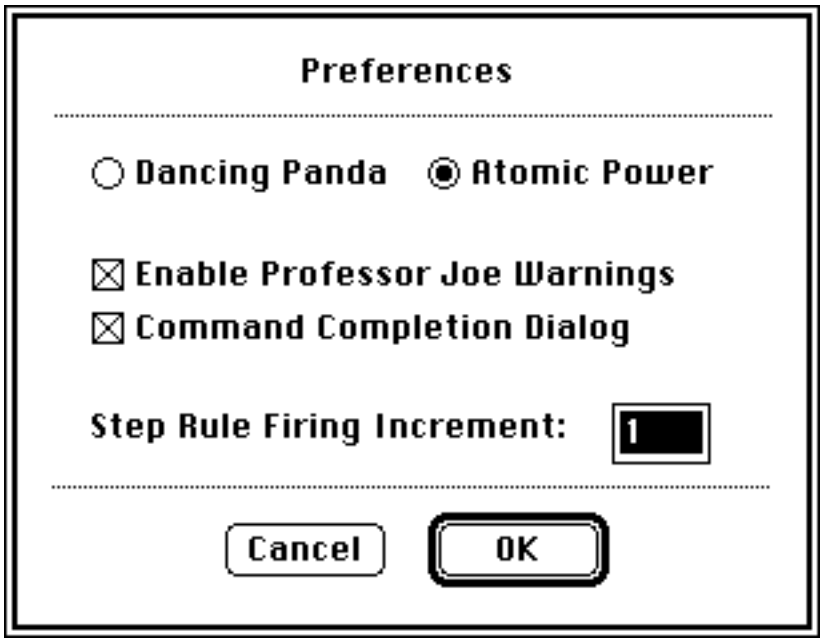
Select a font and font size from the popup menus on the left of the dialog. The sentence at the bottom of the dialog will be displayed using the selected font. Click the **OK** button to change the font used in the currently active editing buffer. Click the **Cancel** button to retain the current font setting. This command is not available when CLIPS is executing.

3.3.9 Clear Window

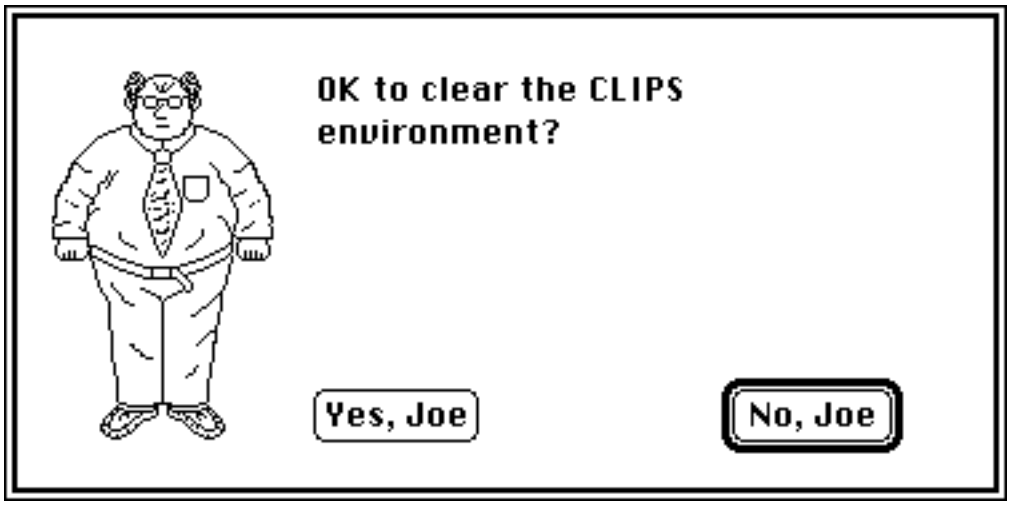
This command clears all of the text in the dialog window. From the CLIPS command prompt, the command **clear-window** (which takes no arguments) will also clear all of the text in the dialog window. This command is available when CLIPS is executing.

3.3.10 Preferences...

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



The **Dancing Panda** and **Atomic Power** radio buttons determine which cursor CLIPS displays when it is busy. The default for this option is **Atomic Power**. From the CLIPS command prompt, the command **set-busy-cursor** can be used to select which cursor is used (pass the symbol *atomic-power* for the atomic power cursor and the symbol *dancing-panda* for the dancing power cursor). The **Enable Professor Joe Warnings** option determines whether certain warnings are issued. The default for this option is on. When Professor Joe 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 the Professor Joe feature on will display the following dialog box:



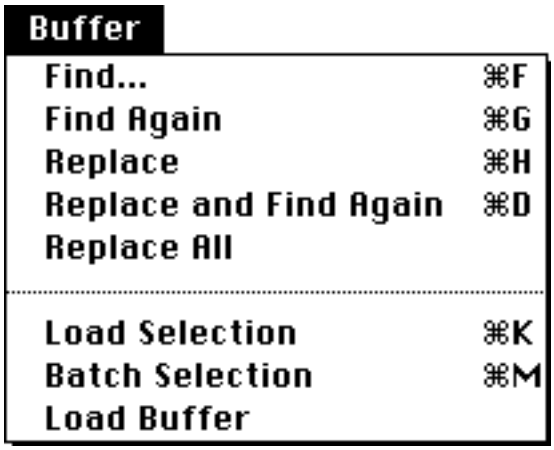
Professor Joe Warnings will only be issued for commands chosen from the menu bar. For example, typing a clear command directly into the dialog window and executing it will not display a warning dialog box regardless of whether Professor Joe Warnings are enabled. From the CLIPS command prompt, the command **set-joe-warnings** can be used to enable and disable Professor Joe Warnings (pass the symbol *TRUE* to enable warnings and the symbol *FALSE* to disable warnings).

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 completions 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.

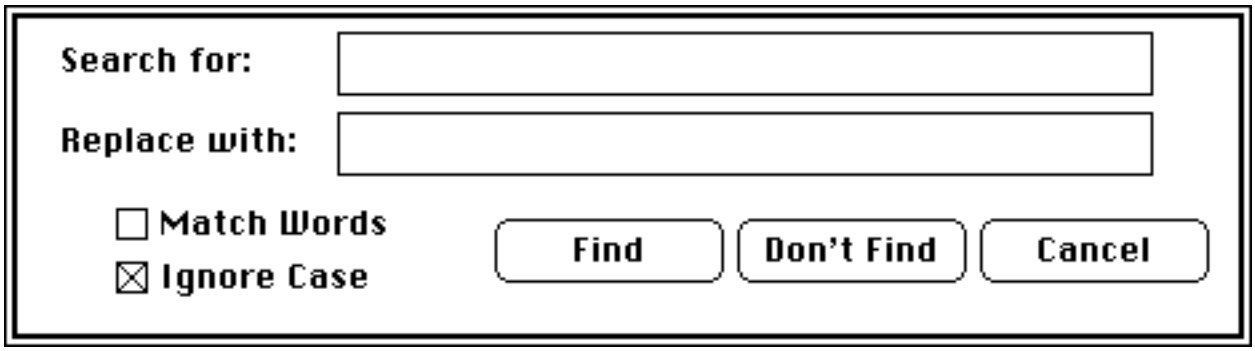
This command is available when CLIPS is executing.

3.4 THE BUFFER MENU



3.4.1 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.



Two other options can be set in the search dialog box. The **Match Words** option will only allow a string search to match whole words; that is, the string “at” would not be bound in the string “match”. The default for this setting is off. 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 default for this setting is on.

Once search options have been set, one of three search dialog buttons can be pressed. The **Find** button will search for the first occurrence of the **Search For** string in the active edit window starting at the end of the current selection and preceding to the end of the buffer. The **Don’t Find** button saves the search and replace strings and options but does not initiate a search. The **Cancel** button aborts the find command.

The search and replace strings and options are restored to the values they had before the dialog box was entered.

This command is not available when CLIPS is executing.

3.4.2 Find Again (⌘G)

This command searches for the next occurrence of the **Search For** string starting at the end of the current selection and preceding to the end of the buffer. This command is not available when CLIPS is executing.

3.4.3 Replace (⌘H)

This command will replace the current selection in the active edit window with the **Replace With** string. If no selection exists, the **Replace With** string will be inserted at the current insertion point in the active edit window. The **Replace With** string may be left empty to replace text with nothing. This command is not available when CLIPS is executing.

3.4.4 Replace and Find Again (⌘D)

This command is equivalent to executing a **Replace** command followed by a **Find Again** command. This command is not available when CLIPS is executing.

3.4.5 Replace All

This command is equivalent to executing a **Find Again** command followed by a **Replace** command repeatedly until the **Find Again** command does not find the **Search For** string. This command is not available when CLIPS is executing.

3.4.6 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). This command is not available when CLIPS is executing.

3.4.7 Batch Selection (⌘M)

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.

3.4.8 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.

3.5 THE EXECUTION MENU

3.5.1 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 dialog window and executed. If Professor Joe Warnings are enabled and 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.

3.5.2 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 dialog window and executed. During execution, this menu command is changed to **Halt** and can be selected to halt

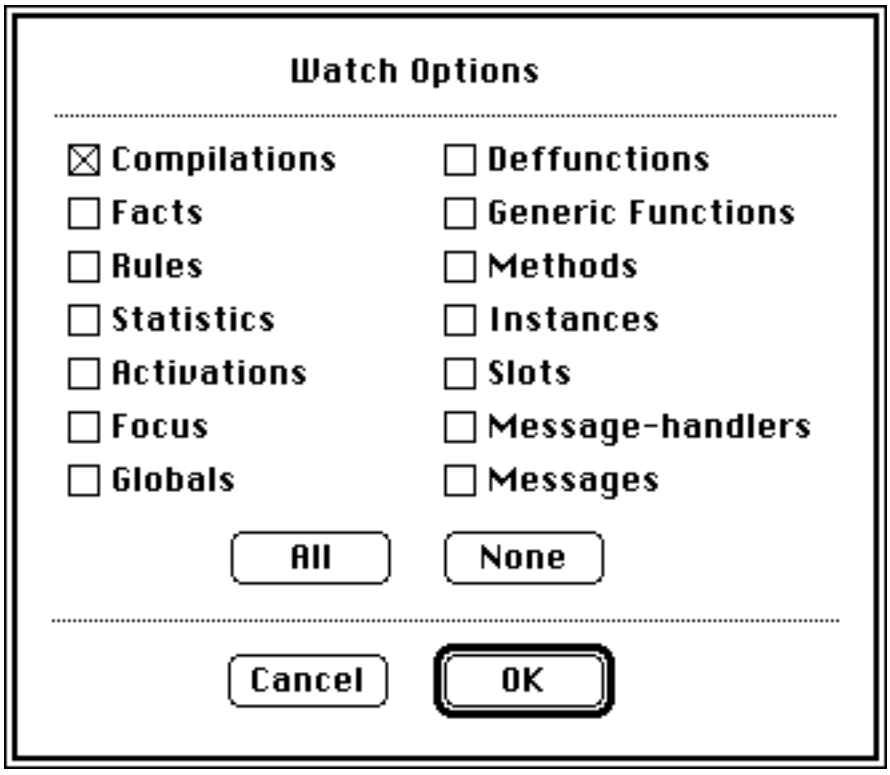
execution (as if a command period had been entered). Holding the shift key down while selecting this menu item is equivalent to entering command-shift-period. For an explanation of the different methods of halting execution refer to the introduction at the beginning of this section.

3.5.3 Step (⌘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.

3.5.4 Watch...

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



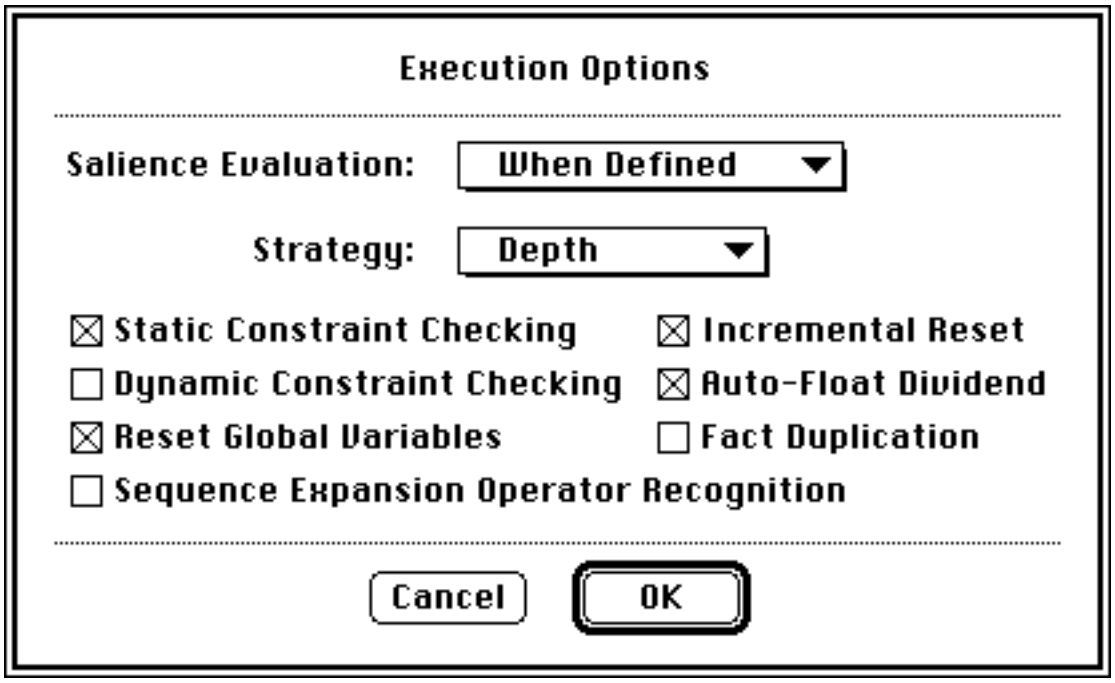
Watch items can be enabled or disabled by clicking the appropriate check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item

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.

3.5.5 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.

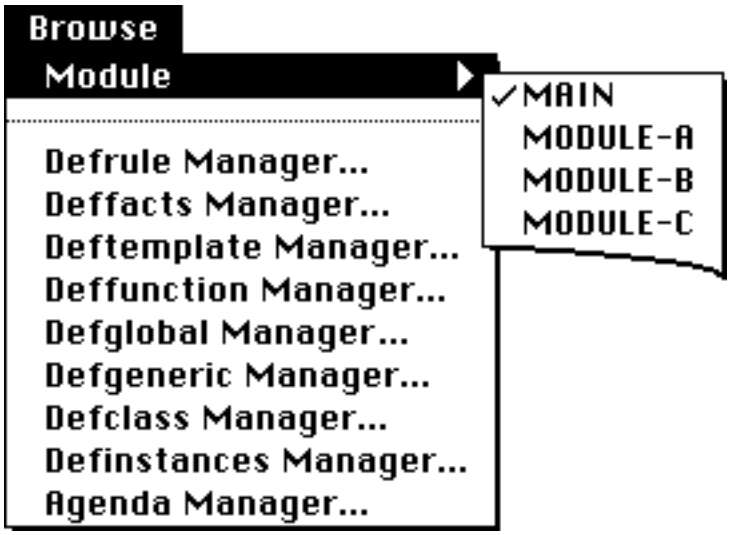
This command is available when CLIPS is executing.

3.5.6 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 Professor Joe 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.

3.6 THE BROWSE MENU

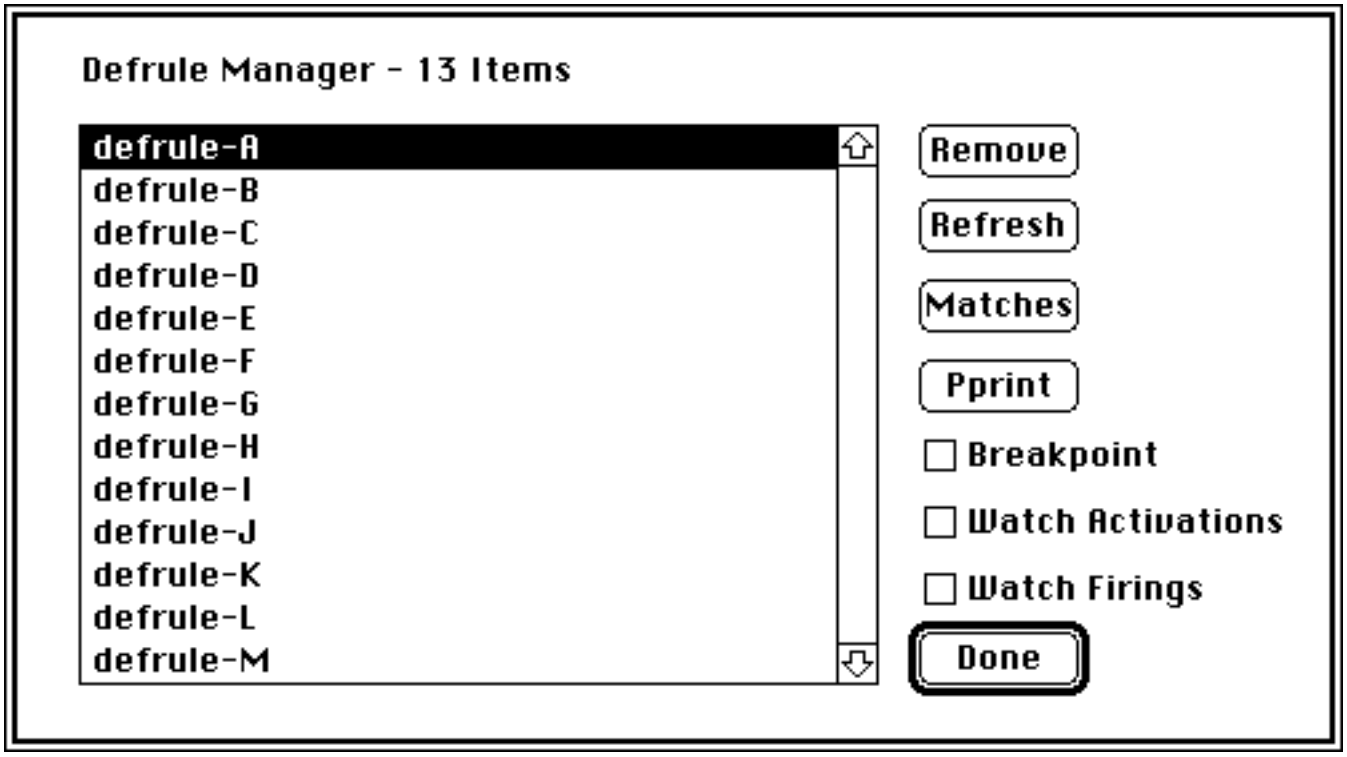
3.6.1 The Module Menu



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.

3.6.2 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.



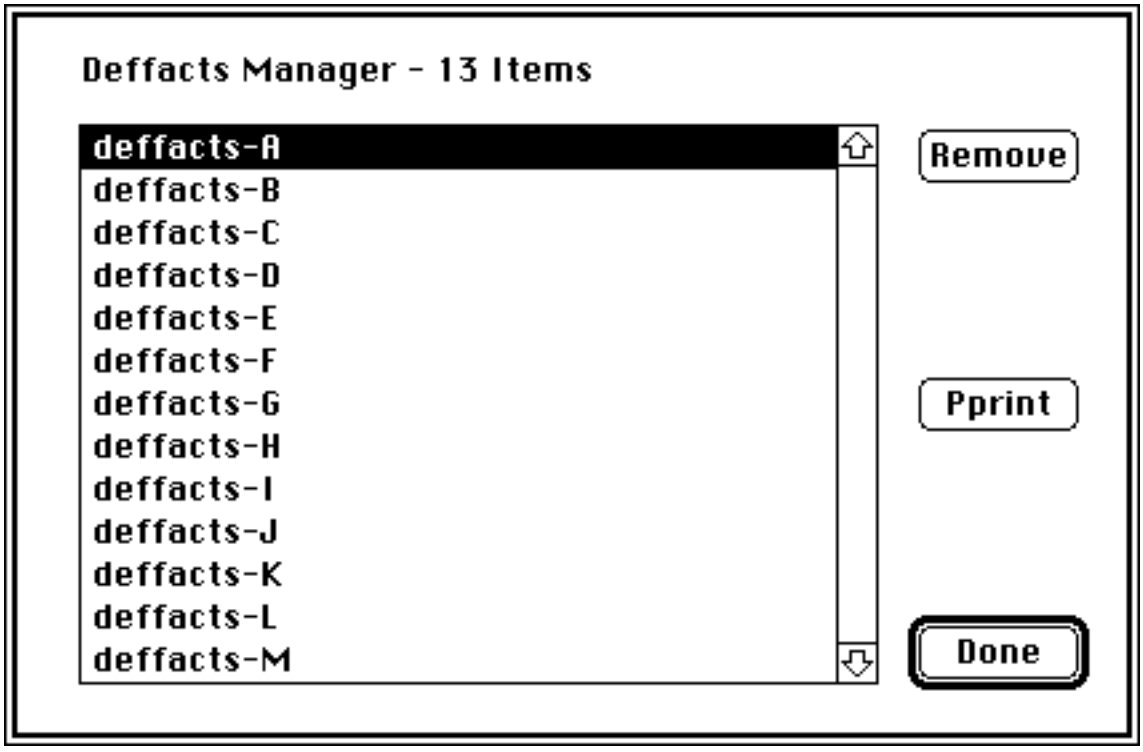
The **Remove** button will remove the currently selected defrule. Pressing this button is equivalent to issuing an **undefrule** command. The **Refresh** button will refresh the currently selected defrule. Pressing this button is equivalent to a **refresh** command. The **Matches** command will list 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.

3.6.3 Deffacts Manager...

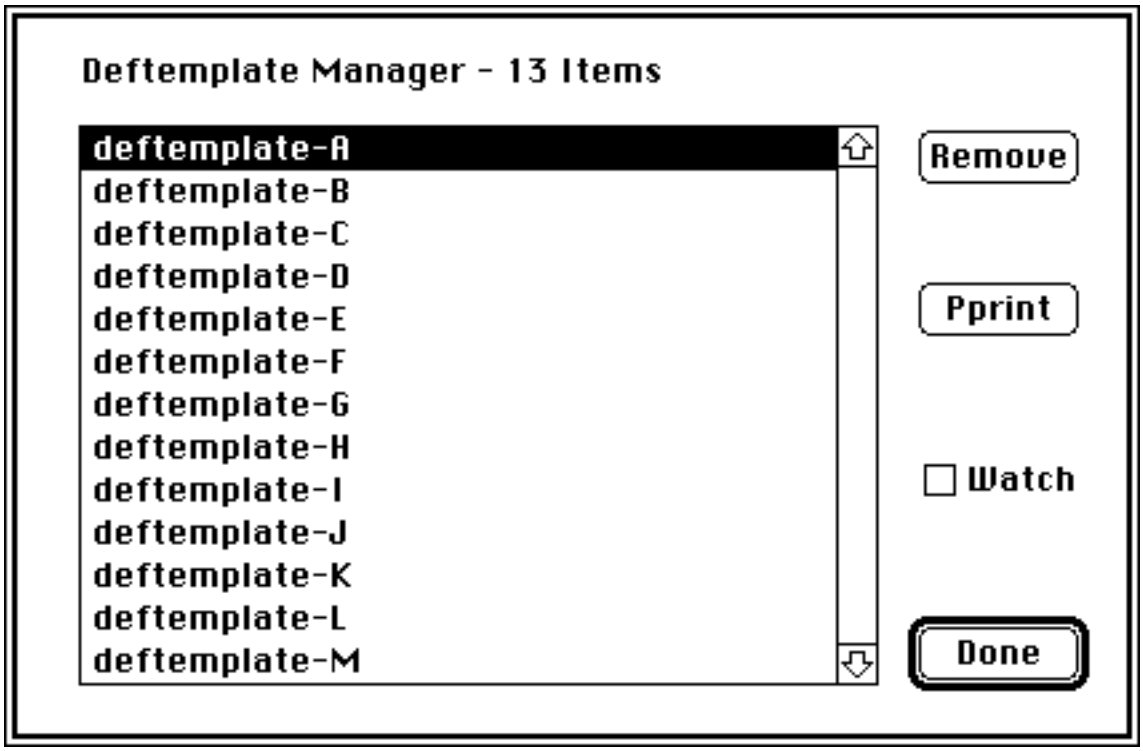
This command displays a dialog box which allows the deffacts in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a deffacts selected from the list of deffacts currently in the knowledge base.



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.

3.6.4 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.



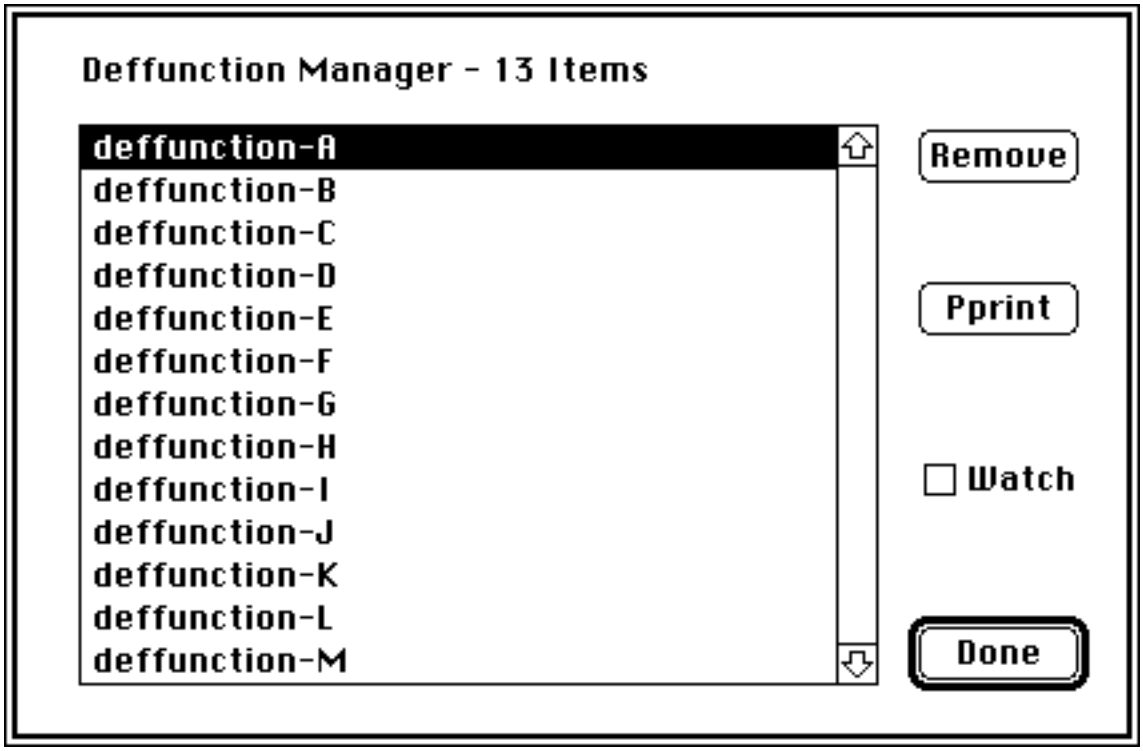
The **Remove** button will remove the currently selected deftemplate. Pressing this button is equivalent to an **undeftemplate** command. The **Pprint** button will pretty print the currently selected deftemplate. Pressing this button is equivalent to a **ppdeftemplate** command. The **Pprint** button 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.

3.6.5 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.



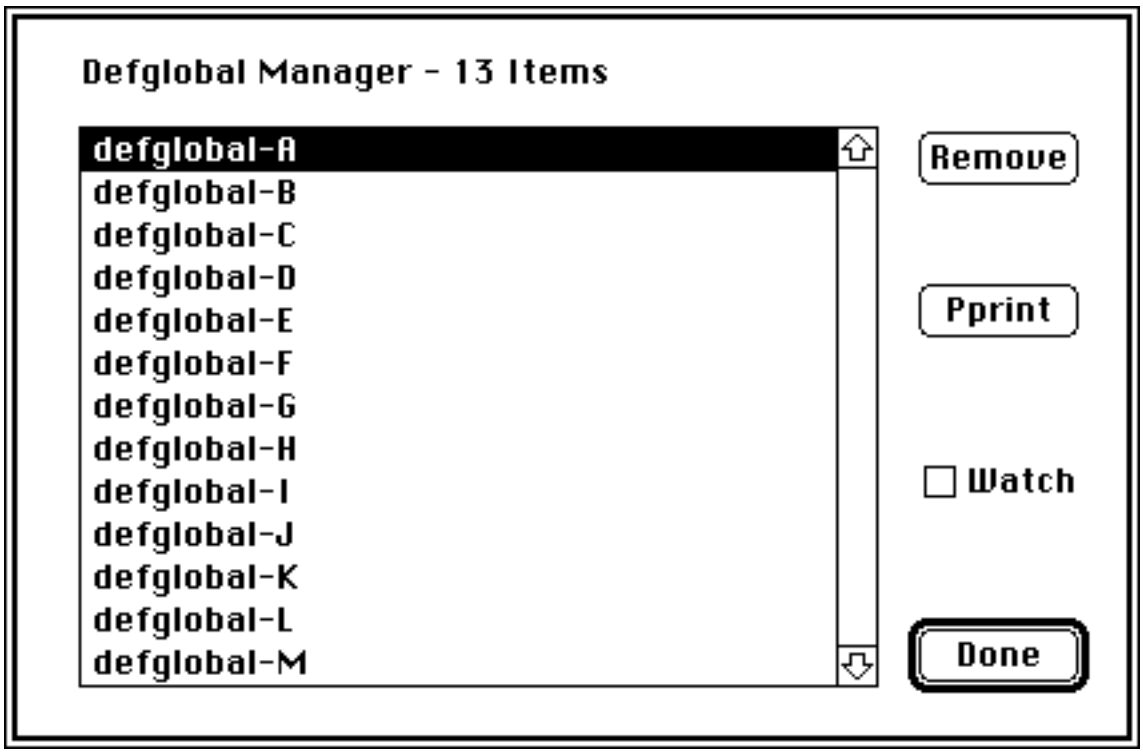
The **Remove** button will remove the currently selected deffunction. Pressing this button is equivalent to an **undeffunction** command. The **Pprint** button will pretty print the currently selected deffunction. Pressing this button is equivalent to a **ppdeffunction** command. The **Pprint** button 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.

3.6.6 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.



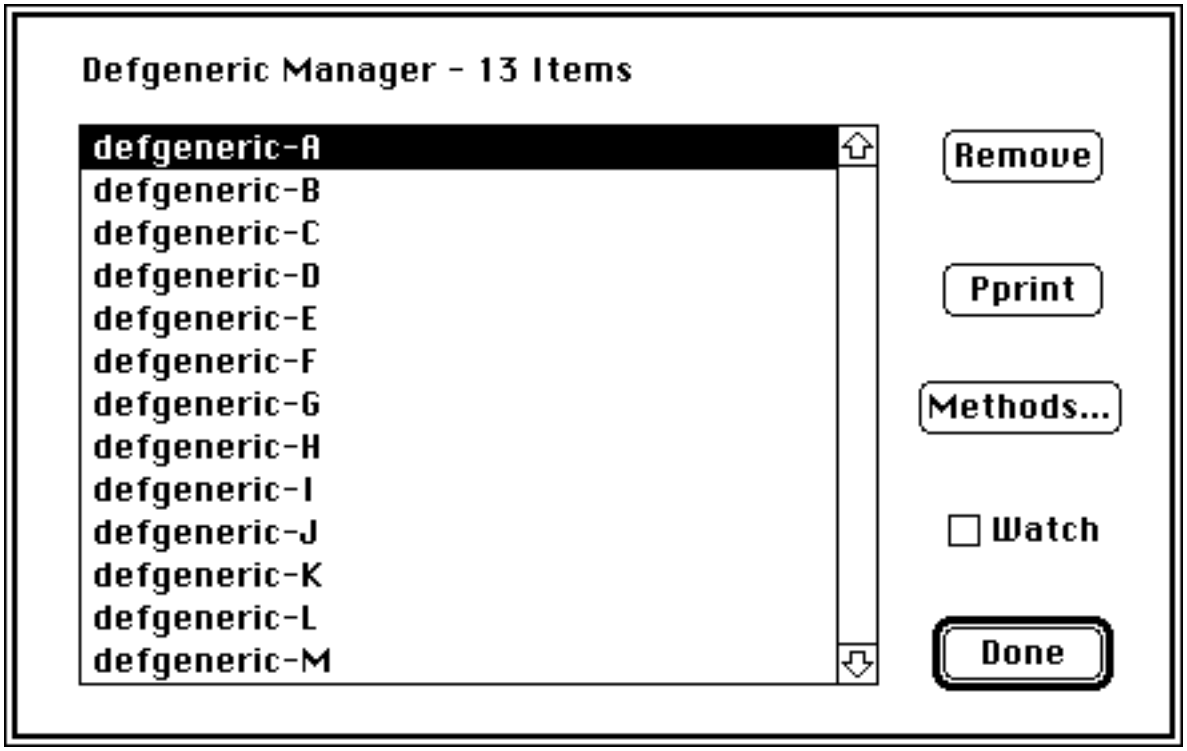
The **Remove** button will remove the currently selected defglobal. Pressing this button is equivalent to an **undefglobal** command. The **Pprint** button will pretty print the currently selected defglobal. Pressing this button is equivalent to a **ppdefglobal** command. The **Pprint** button echoes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

The **Watch** check box is used to enable or disable the messages that are printed when the value of a global variable is changed. If the check box for a defglobal is checked, then value change messages will be printed.

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.

3.6.7 Defgeneric Manager...

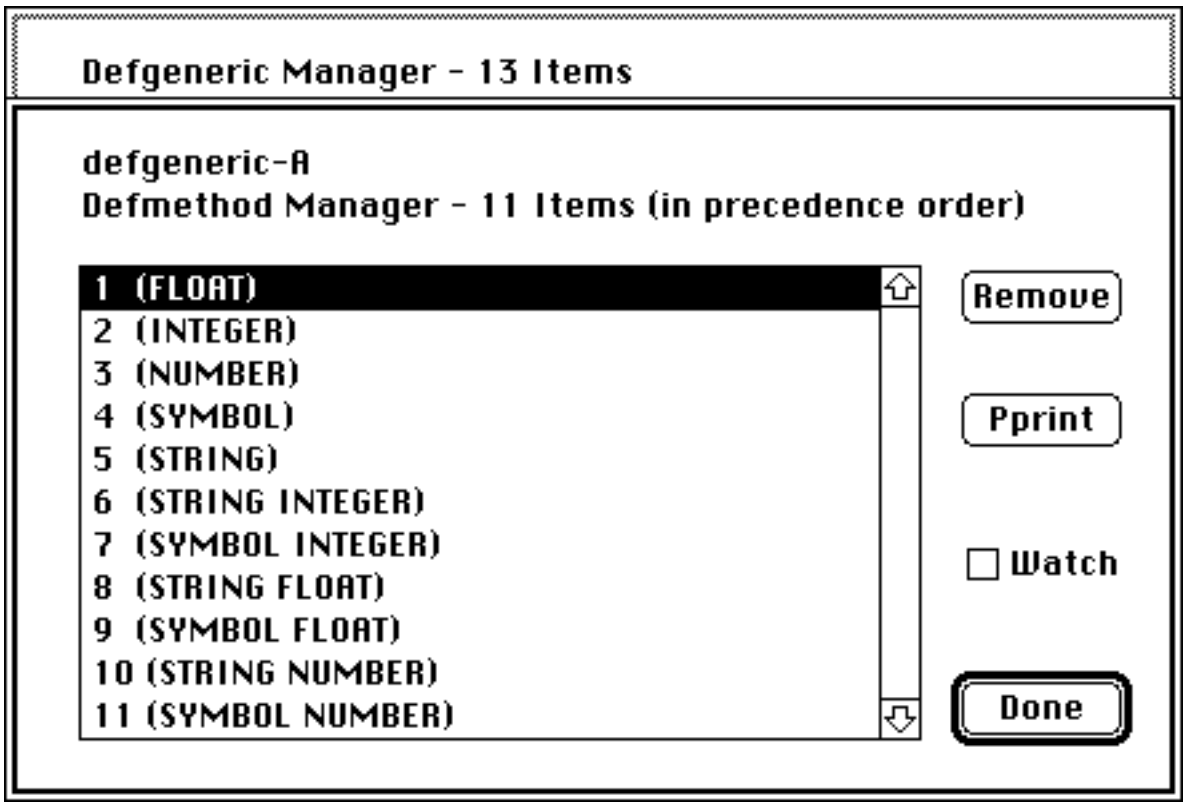
This command displays a dialog box which allows the defgenerics in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defgeneric selected from the list of defgenerics currently in the knowledge base. Any number of operations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

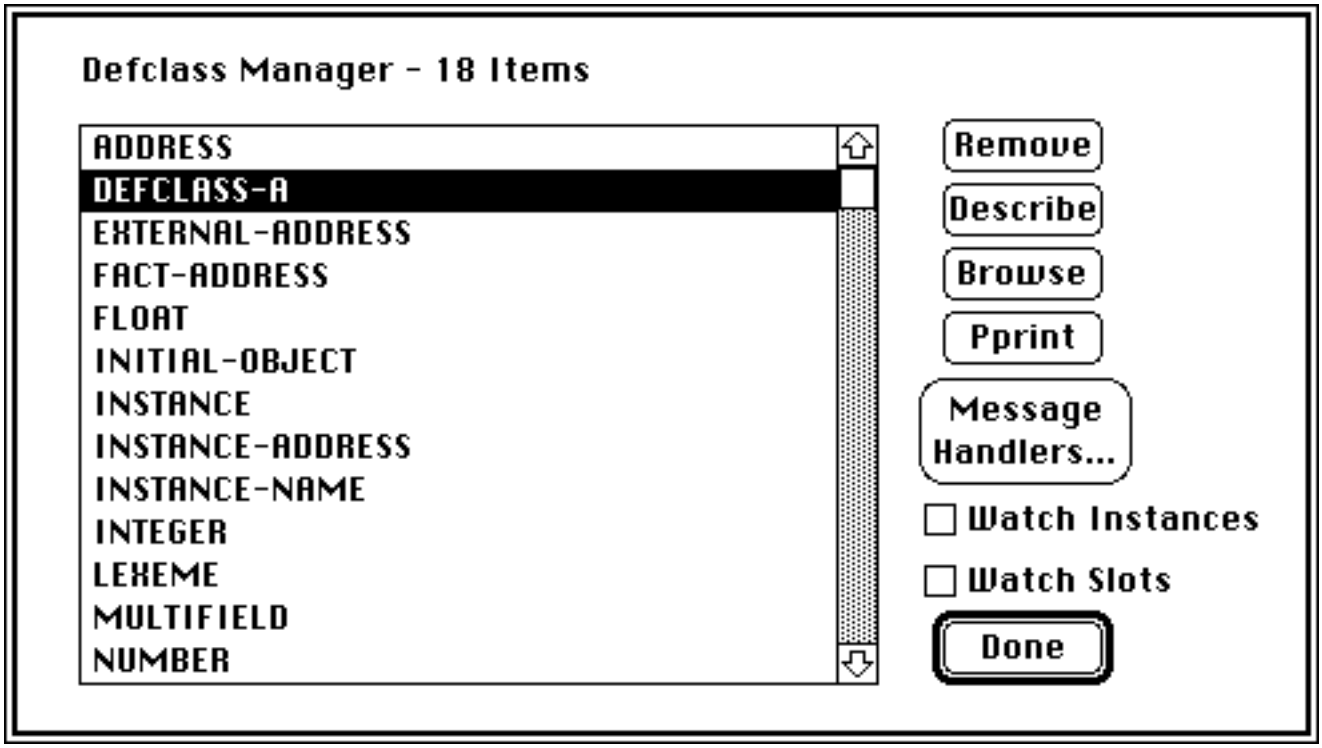
The **Watch** check box is used to enable or disable the messages that are printed when a specific method of a generic function begins and ends execution. If the check box for a defmethod is checked, then execution messages for the method will be printed.

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.

3.6.8 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.

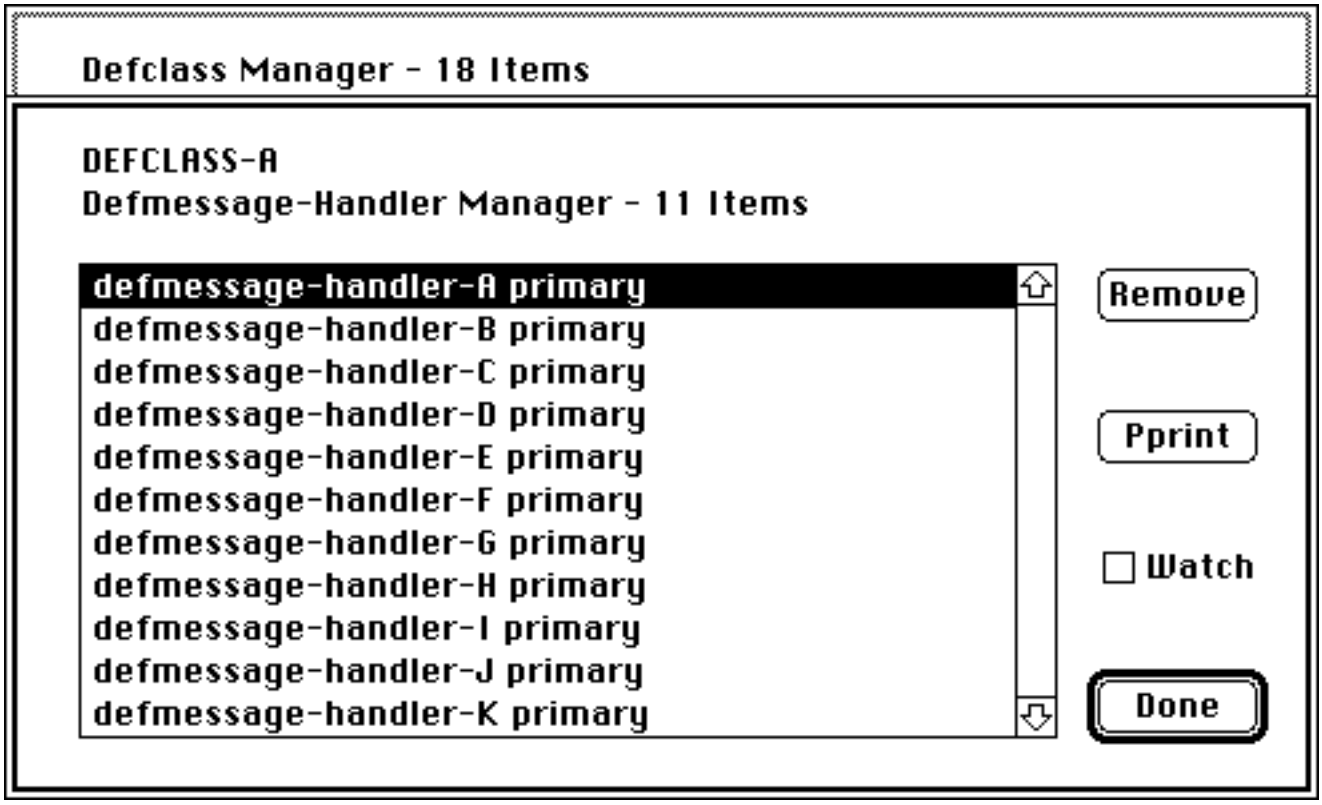


The **Remove** button will remove the currently selected defclass. Pressing this button is equivalent to an **undefclass** command. The **Describe** button will describe the currently selected defclass. Pressing this button is equivalent to a **describe-class** command. The **Browse** command displays 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.

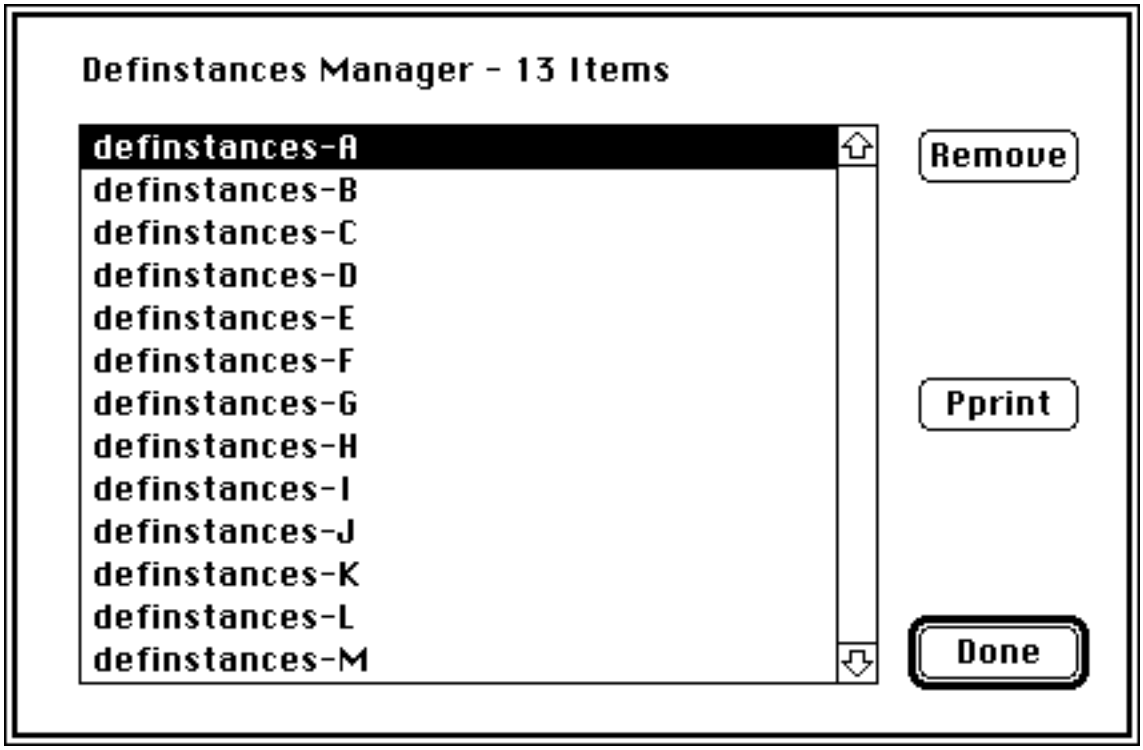
The **Watch** check box is used to enable or disable the messages that are printed when a message-handler begins and ends execution. If the check box for a defmessage-handler is checked, then execution messages for the message-handler will be printed.

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.

3.6.9 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

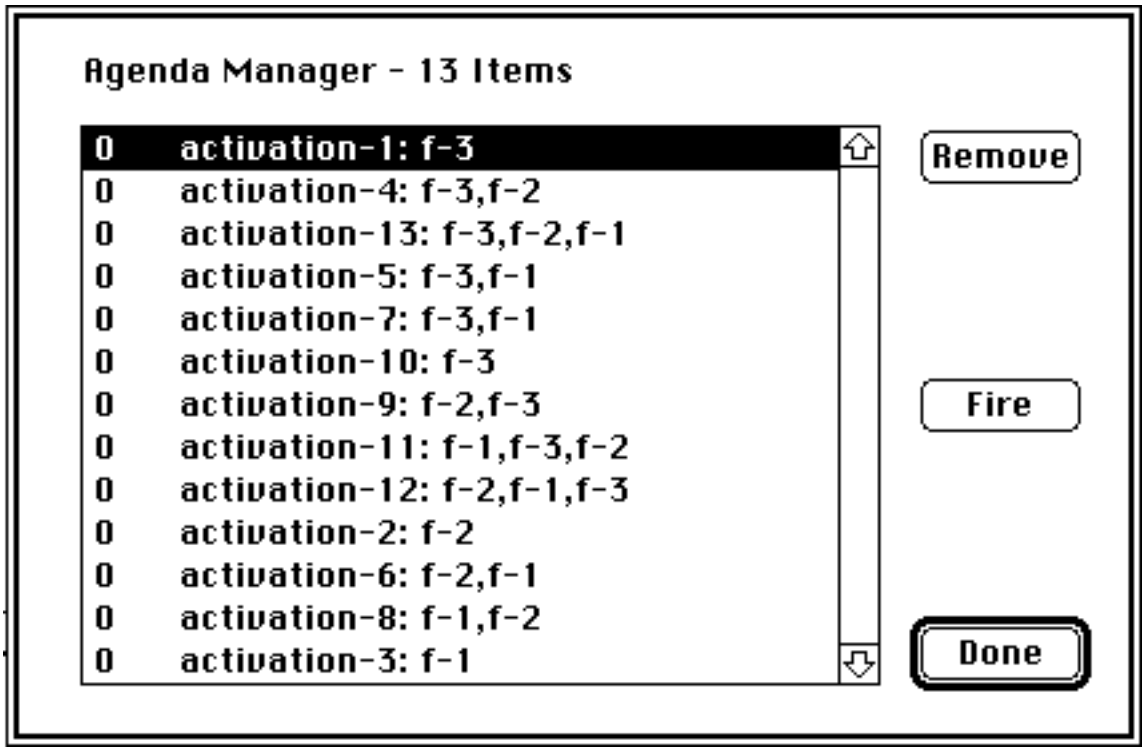
allows an operation to be performed on a definstances selected from the list of definstances currently in the knowledge base.



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.

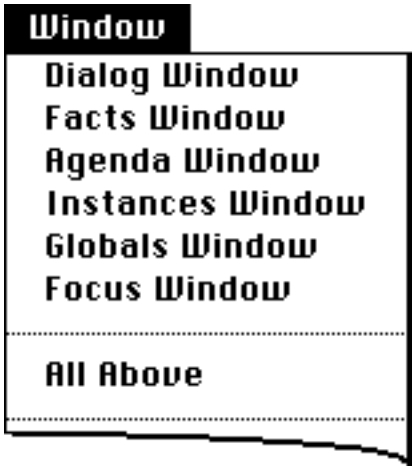
3.6.10 Agenda Manager...

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.



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.

3.7 THE WINDOW MENU



3.7.1 Dialog Window

This command selects the **Dialog Window** and brings it to the front. 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.

3.7.2 Facts Window

This command selects the **Facts Window** and brings it to the front. If the **Facts Window** does not exist, it will be created. The **Facts Window** displays the list of facts in the fact-list. 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.

3.7.3 Agenda Window

This command selects the **Agenda Window** and brings it to the front. If the **Agenda Window** does not exist, it will be created. The **Agenda Window** displays the list of activations currently on the agenda. 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.

3.7.4 Instances Window

This command selects the **Instances Window** and brings it to the front. If the **Instances Window** does not exist, it will be created. 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 is the frontmost window. This command is available when CLIPS is executing.

3.7.5 Globals Window

This command selects the **Globals Window** and brings it to the front. If the **Globals Window** does not exist, it will be created. 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 is the frontmost window. This command is available when CLIPS is executing.

3.7.6 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.

3.7.7 All Above

This command brings the **Dialog Window**, **Facts Window**, **Instances Window**, **Globals Window**, and **Focus Window** in front of all other windows (creating them if they do not already exist). The **Dialog Window** is selected as the active window. This command is available when CLIPS is executing.

3.7.8 Titled and Untitled Edit Windows

This command brings the selected edit window to the front. All edit windows appear below the dotted line in the menu. 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.

3.8 CREATING THE MACINTOSH EXECUTABLES

The following four subsections describe how to create generic CLIPS and Macintosh CLIPS interface executables on the Macintosh using the THINK C v5.0 compiler and MPW 3.2 C. Note that some familiarity with building applications using THINK C and MPW C 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 Macintosh executables.

3.8.1 Building the CLIPS Generic Executable Using THINK C

The following steps describe how to create the generic executable version of CLIPS using THINK C v5.0:

1) Create a folder named **THINK C CLIPS**. Place the file **CLIPS 6.0G.π** (included on the CLIPS Macintosh distribution disks) into this folder.

2) Create a folder named **CLIPS Source** and place it in the **THINK C CLIPS** folder. Copy all of the CLIPS source code (excluding the source code for the machine specific interfaces) into the **CLIPS Source** folder.

3) Launch the THINK C application by double clicking on the **CLIPS 6.0G.π** file.

4) Edit the **setup.h** file. Set the compiler directive flags MAC_MPW to 0 and WINDOW_INTERFACE to 0. Set the compiler directive flag MAC_TC to 1.

5) Select **Build Application...** in the **Project** menu and save the executable version under the desired name and location (the default name for the executable is **CLIPS 6.0G**).

3.8.2 Building the CLIPS Interface Executable Using THINK C

The following steps describe how to create the Macintosh machine specific version of CLIPS using THINK C v5.0:

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

alpha.h	clipsmnu.h	clipsstw.h	common.h
cooldlog.h	displayw.h	dlogmngr.h	editdlog.h
editmenu.h	editw.h	execdlog.h	factdlog.h
filemenu.h	fontpick.h	gnrcdlog.h	interface.h
macinit.h	mainloop.h	menucmds.h	menuhndl.h
prcddlog.h	print.h	ruledlog.h	scrap.h
search.h	statusw.h	stnrdmac.h	undo.h
window.h			

alpha.c	clipsmnu.c	clipsstw.c	cooldlog.c
displayw.c	dlogmngr.c	editdlog.c	editmenu.c
editw.c	execdlog.c	factdlog.c	filemenu.c
fontpick.c	gnrcdlog.c	macfuncs.c	macinit.c
macmain.c	mainloop.c	menucmds.c	menuhndl.c
prcddlog.c	print.c	ruledlog.c	scrap.c
search.c	statusw.c	stnrdmac.c	undo.c
window.c			

CLIPS 6.0.π.Rsrc

2) Create a folder named **THINK C CLIPS**. Place the files **CLIPS 6.0.π** and **CLIPS 6.0.π.Rsrc** (included on the CLIPS Macintosh distribution disks) into this folder.

3) Create a folder named **CLIPS Source** and place it in the **THINK C CLIPS** folder. Copy all of the CLIPS source code into the **CLIPS Source** folder. Copy all

of the Macintosh interface source code (the .h and .c source listed in step 1) into the **CLIPS Source** folder.

4) Launch the THINK C application by double clicking on the **CLIPS 6.0.π** file.

5) Edit the **setup.h** file. Set the compiler directive flags MAC_MPW to 0 and WINDOW_INTERFACE to 1. Set the compiler directive flag MAC_TC to 1.

6) Select **Build Application...** in the **Project** menu and save the executable version under the desired name and location (the default name for the executable is **CLIPS 6.0**).

3.8.3 Building the CLIPS Generic Executable Using MPW C

The following steps describe how to create the generic executable version of CLIPS using MPW 3.2 C with the version 7.0 Toolbox Interfaces and Libraries:

1) Create a folder named **MPW CLIPS**. Place the file **CLIPS 6.0G.make** (included on the CLIPS Macintosh distribution disks) into this folder.

2) Create a folder named **CLIPS Source** and place it in the **MPW CLIPS** folder. Copy all of the CLIPS source code (excluding the source code for the machine specific interfaces) into the **CLIPS Source** folder.

3) Launch the MPW application.

4) Edit the **setup.h** file. Set the compiler directive flag MAC_MPW to 1. Set the compiler directive flags WINDOW_INTERFACE and MAC_TC to 0.

5) Select **Set Directory...** from the MPW **Directory** menu and select the **MPW CLIPS** folder from the resulting dialog.

6) Select **Build...** from the MPW **Build** menu and enter **CLIPS 6.0G** in response to the resulting dialog. Once the build process is complete, the executable file **CLIPS 6.0G** will be stored in the **MPW CLIPS** folder.

3.8.4 Building the CLIPS Interface Executable Using MPW C

The following steps describe how to create the Macintosh machine specific version of CLIPS using MPW 3.2 C with the version 7.0 Toolbox Interfaces and Libraries:

1) Create a folder named **MPW CLIPS**. Place the files **CLIPS 6.0.make** and **CLIPS 6.0.r** (included on the CLIPS Macintosh distribution disks) into this folder.

2) Create a folder named **CLIPS Source** and place it in the **MPW CLIPS** folder. Copy all of the CLIPS source code into the **CLIPS Source** folder. Copy all of the Macintosh interface source code (the .h and .c source listed in step 1 of section 6.8.2) into the **CLIPS Source** folder. The file **CLIPS6.0.Rsrc** is not needed.

3) Launch the MPW application.

4) Edit the **setup.h** file. Set the compiler directive flags MAC_MPW and WINDOW_INTERFACE to 1. Set the compiler directive flag MAC_TC to 0.

5) Select **Set Directory...** from the MPW **Directory** menu and select the **MPW CLIPS** folder from the resulting dialog.

6) Select **Build...** from the MPW **Build** menu and enter **CLIPS 6.0** in response to the resulting dialog. Once the build process is complete, the executable file **CLIPS 6.0** will be stored in the **MPW CLIPS** folder.

Section 4 - CLIPS X Window Interface

This section provides a brief summary of the CLIPS 6.0 X Window interface (**xclips**) for the SUN workstation. 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).

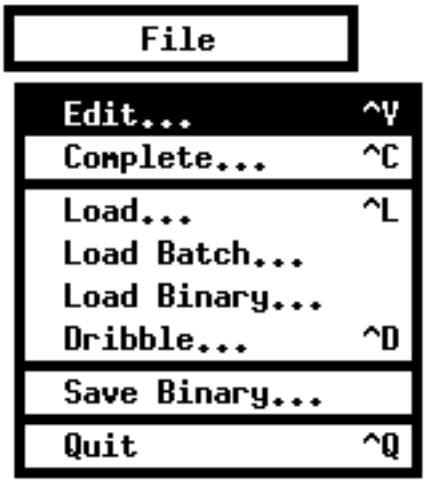
The execution of the current command or CLIPS program can be halted by pressing Ctrl-H

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 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

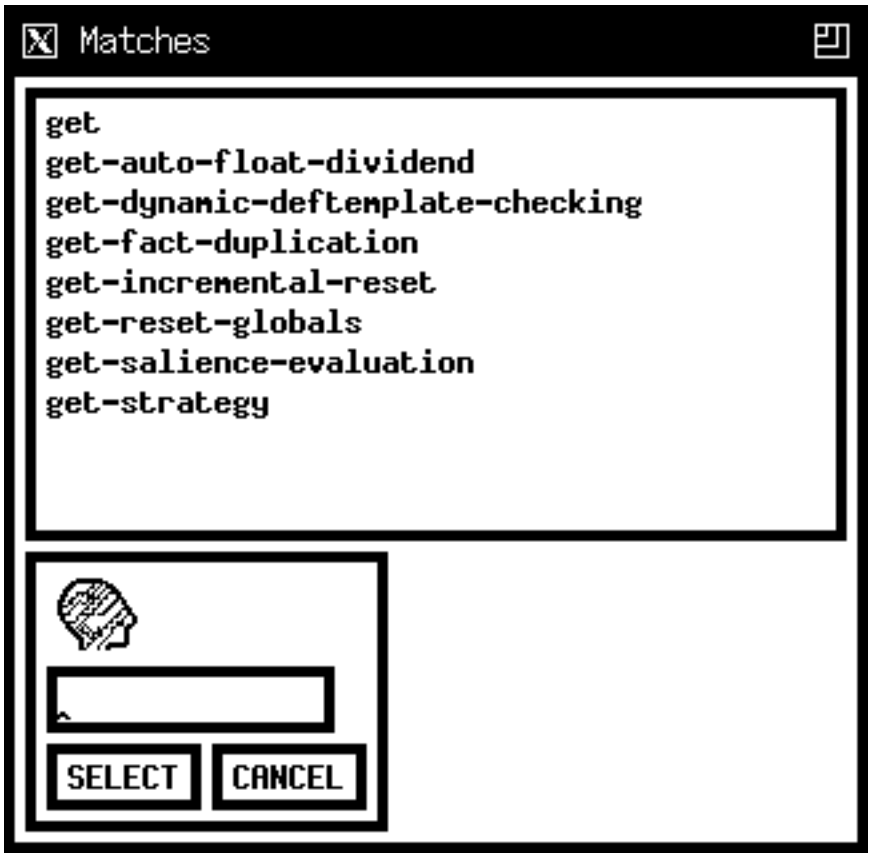


4.2.1 Edit... (^V)

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).

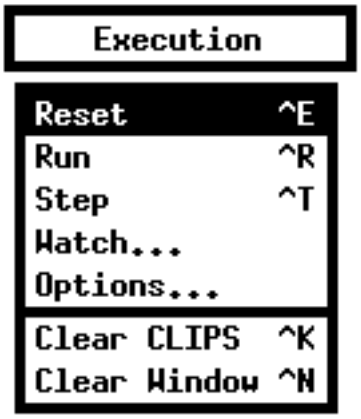
4.2.7 Save Binary...

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



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)

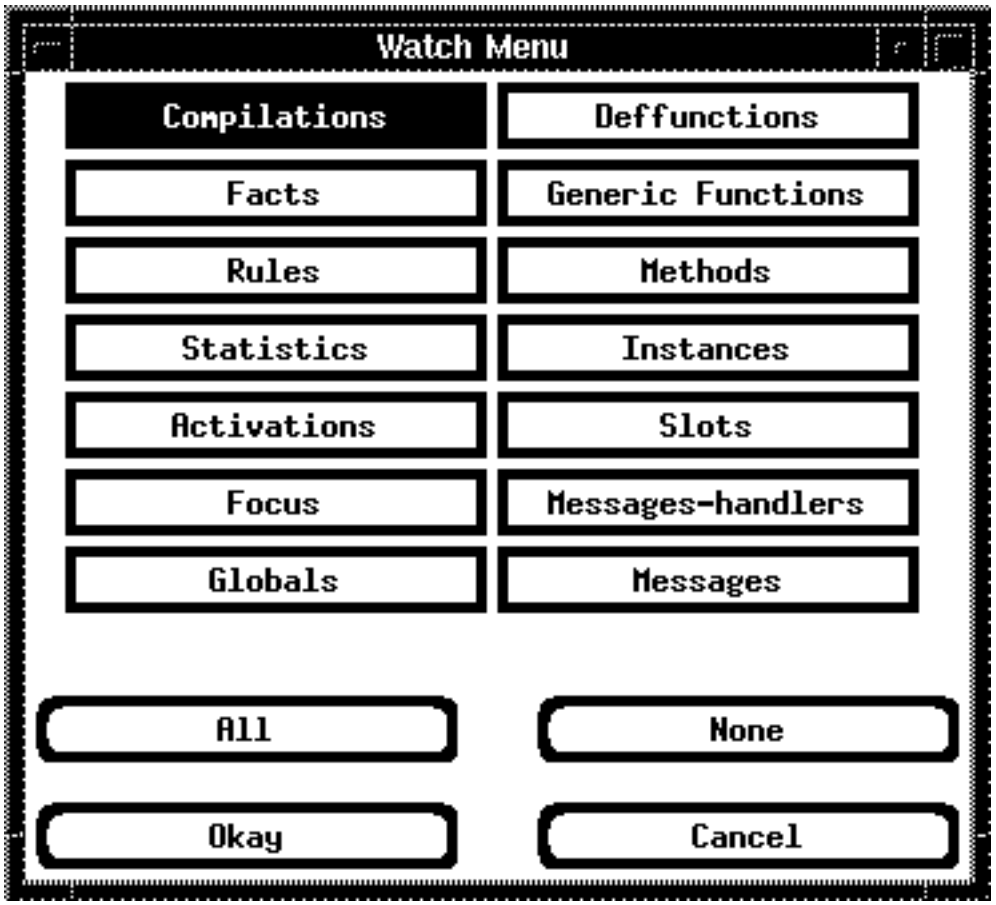
This command is equivalent to the CLIPS command (run). When this command is chosen, the CLIPS command (run) will be echoed to the main window and executed. This command is not available when CLIPS is executing.

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.

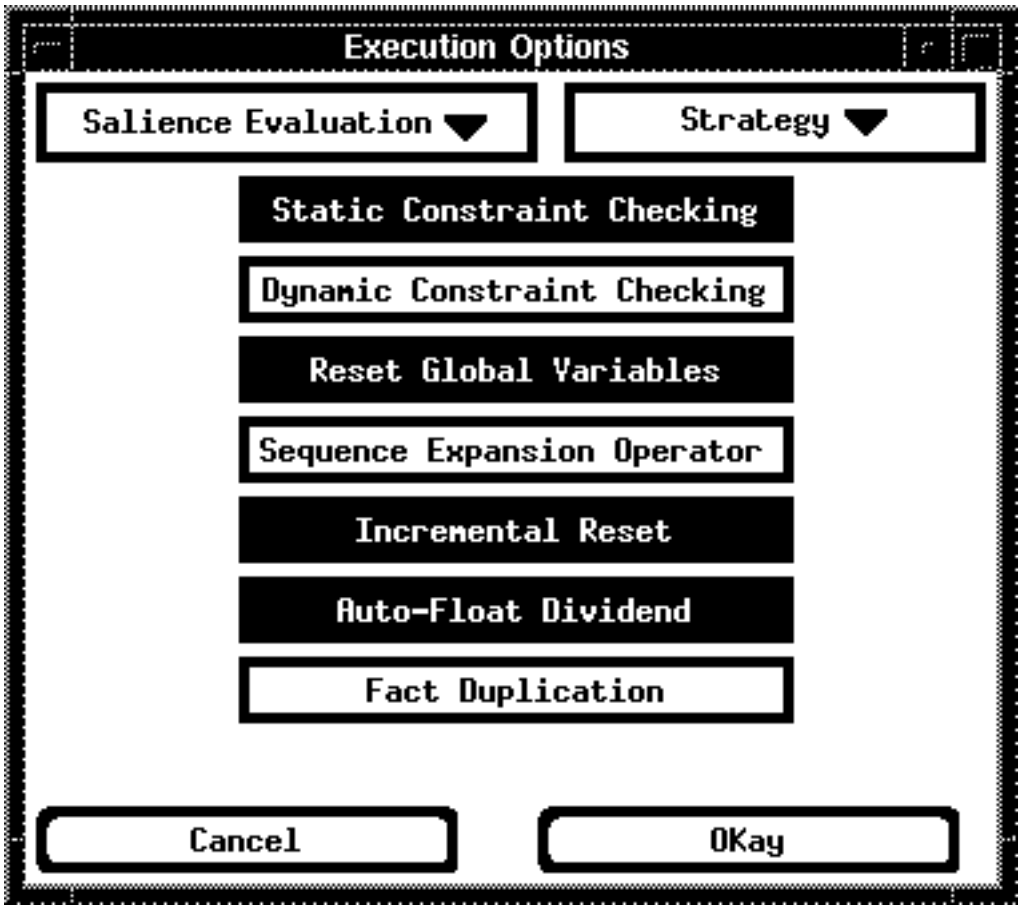


Watch items can be enabled or disabled by clicking its check box. Enabled watch items have a check in their check box. Disabled watch items have no check mark in their check box. Pressing the **All** button checks all of the watch item check boxes. Pressing the **None** button unchecks all of the watch 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.

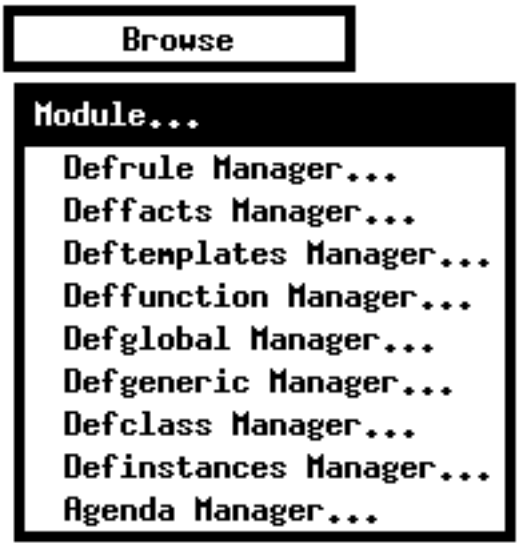
4.3.6 Clear CLIPS (^K)

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)

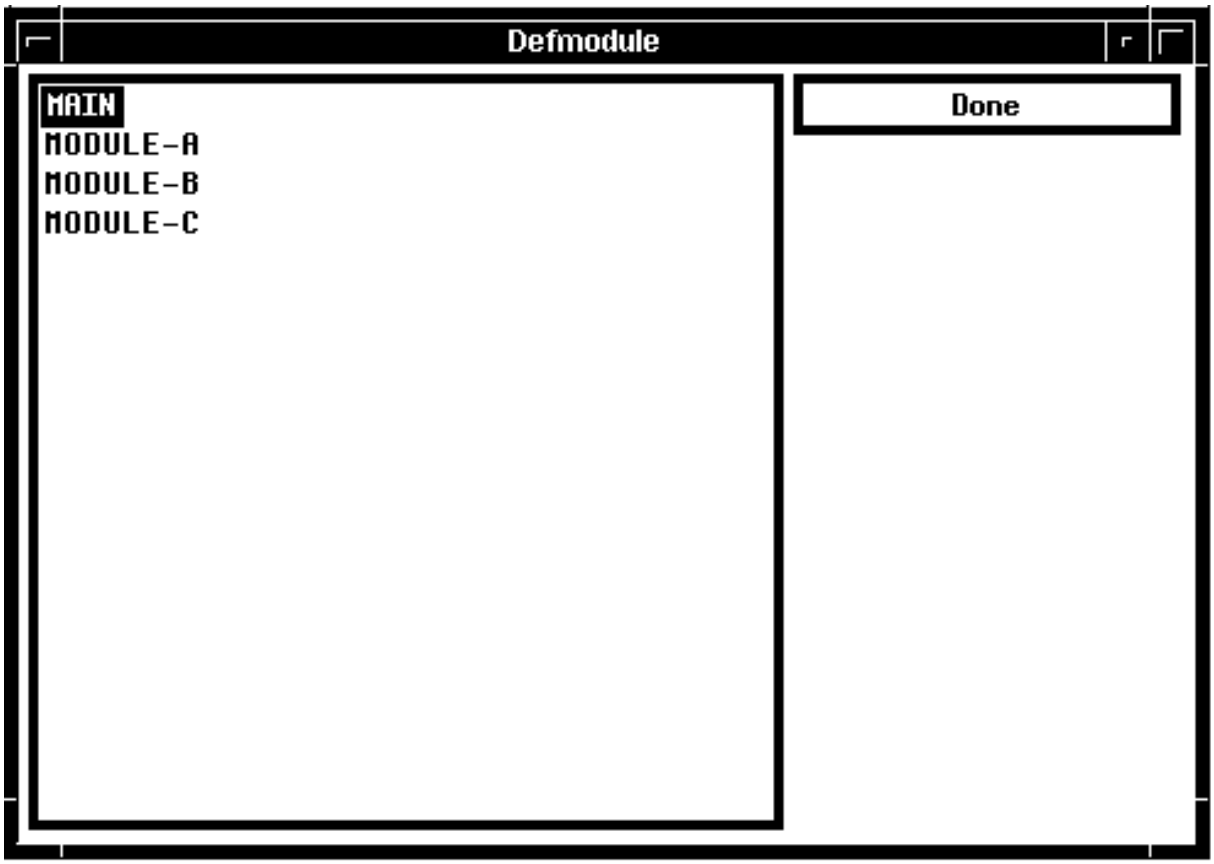
This command clears the dialog window. This command is not available when CLIPS is executing.

4.4 THE BROWSE MENU



4.4.1 Module Menu

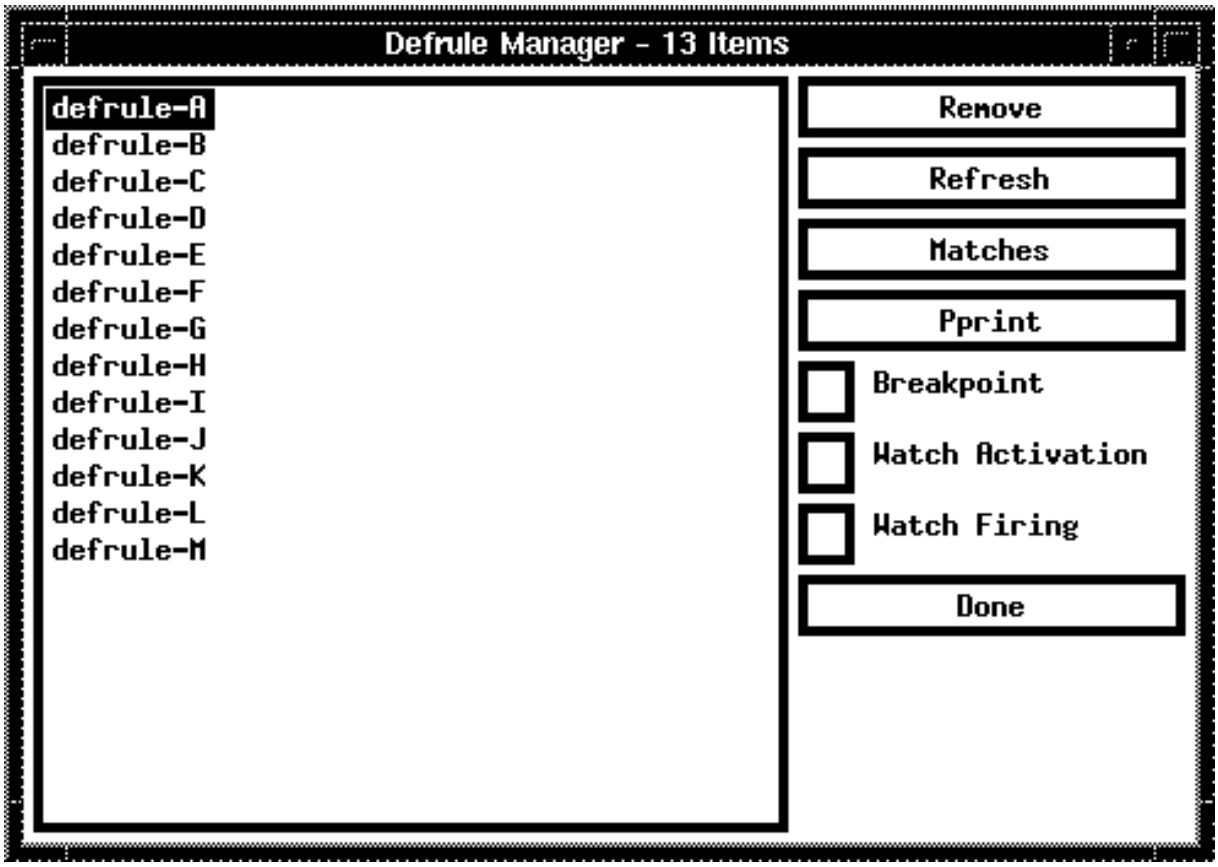
This menu displays a dialog box which contains the list of the defmodules currently defined in the CLIPS environment. A current module is hilighted. Selecting a module from the list changes the current module to the selected item.



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.

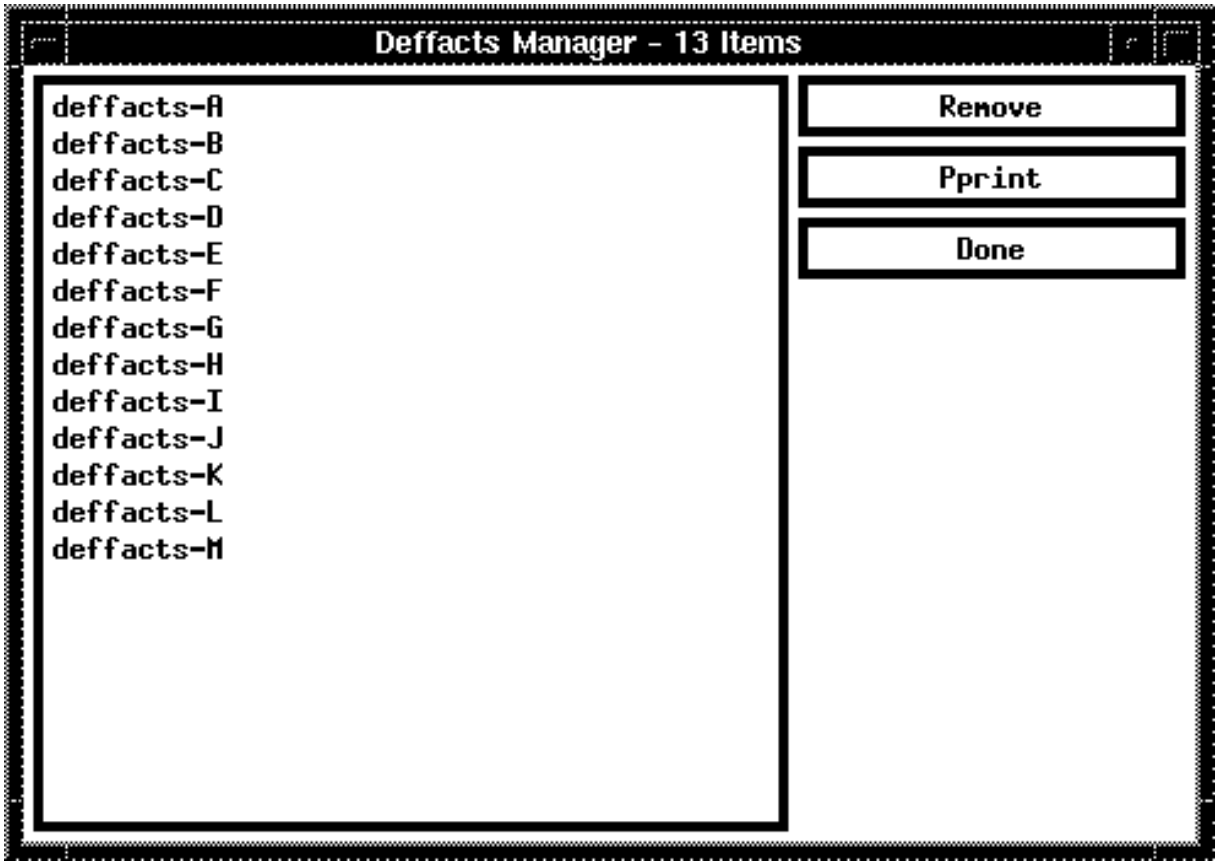
The **Breakpoint** check box is used to enable or disable a breakpoint on the currently selected defrule. Rules with a breakpoint set have a check in their check box while rules without a breakpoint have none. The **Watch Activations** check box is used to enable or disable the messages that are printed 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.

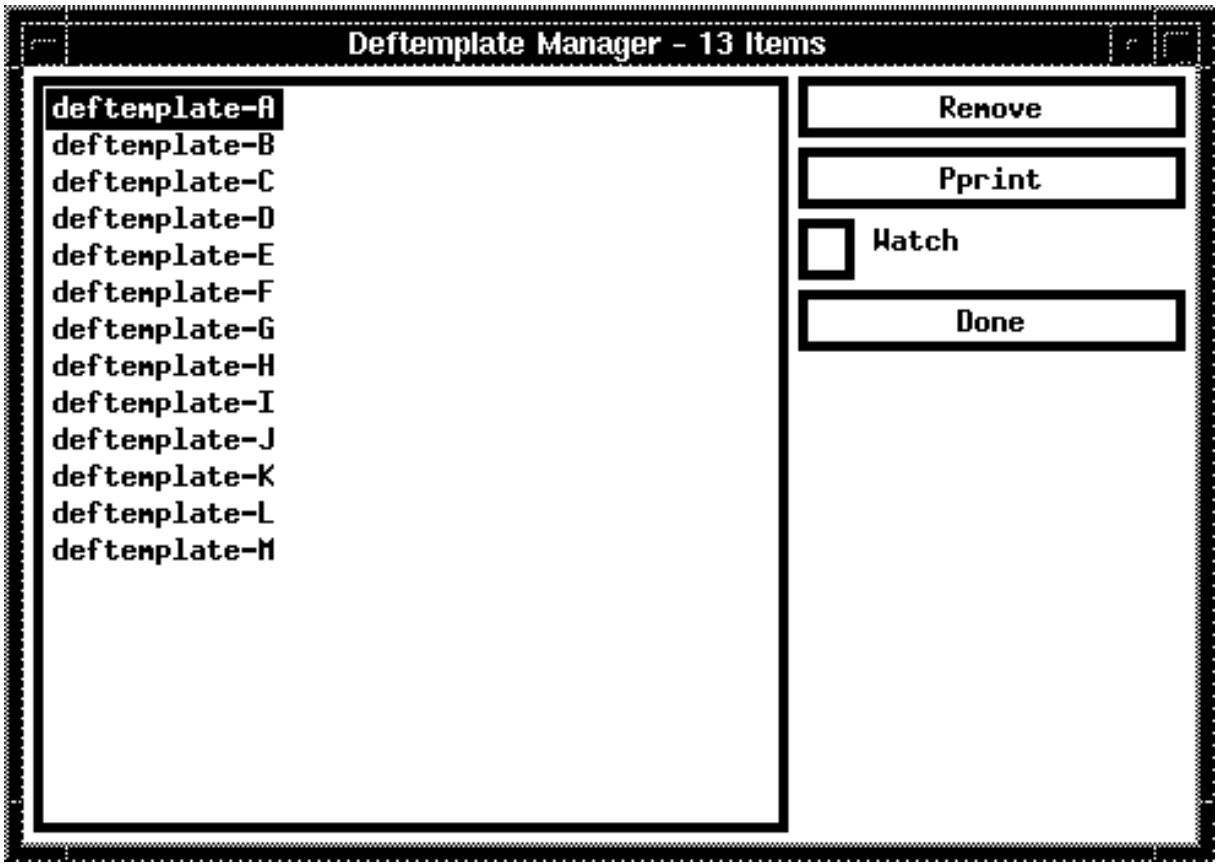


The **Remove** button will remove the currently selected deffacts. Pressing this button is equivalent to an **undeffacts** command The **Pprint** button will pretty print the currently selected deffacts. Pressing this button is equivalent to a **ppdeffacts** command. The **Pprint** button echoes the 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.



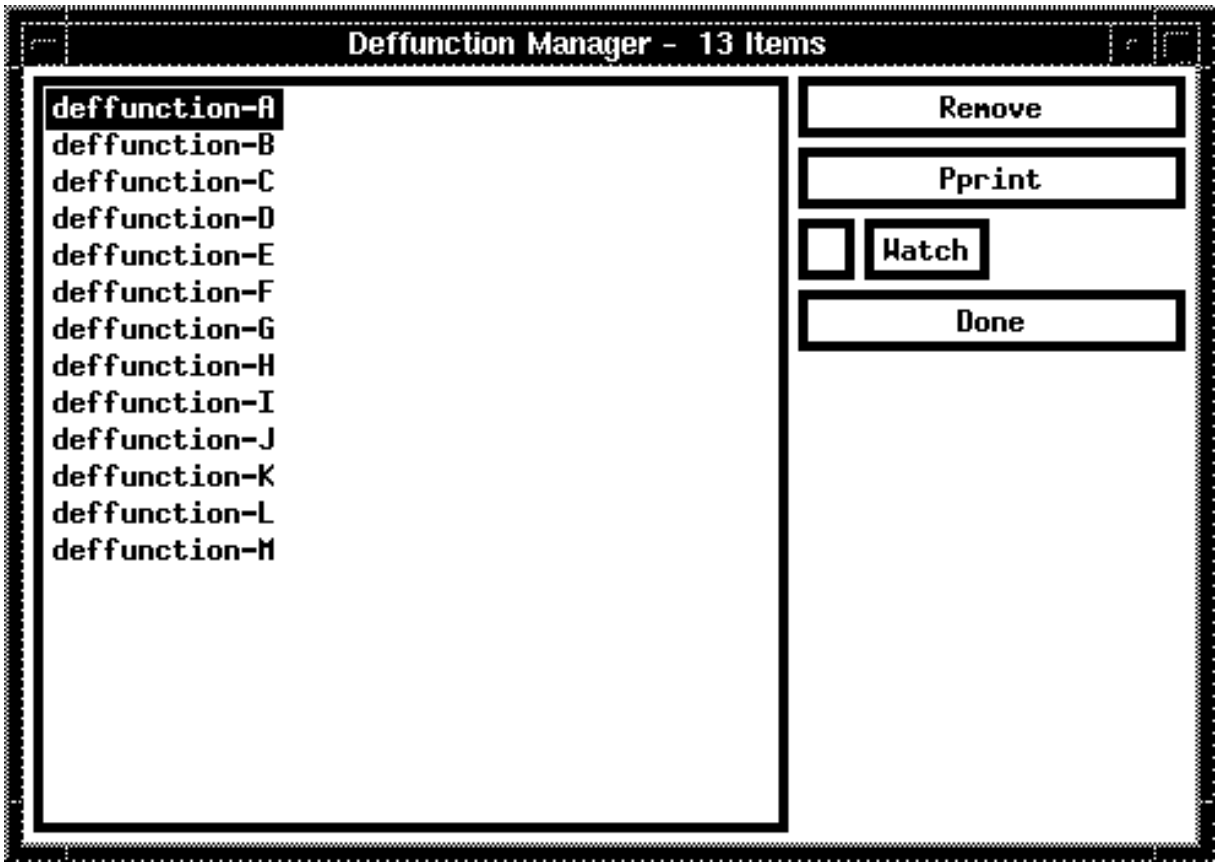
The **Remove** button will remove the currently selected deftemplate. Pressing this button is equivalent to an **undeftemplate** command. The **Pprint** button will pretty print the currently selected deftemplate. Pressing this button is equivalent to a **ppdeftemplate** command. The **Pprint** button 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.



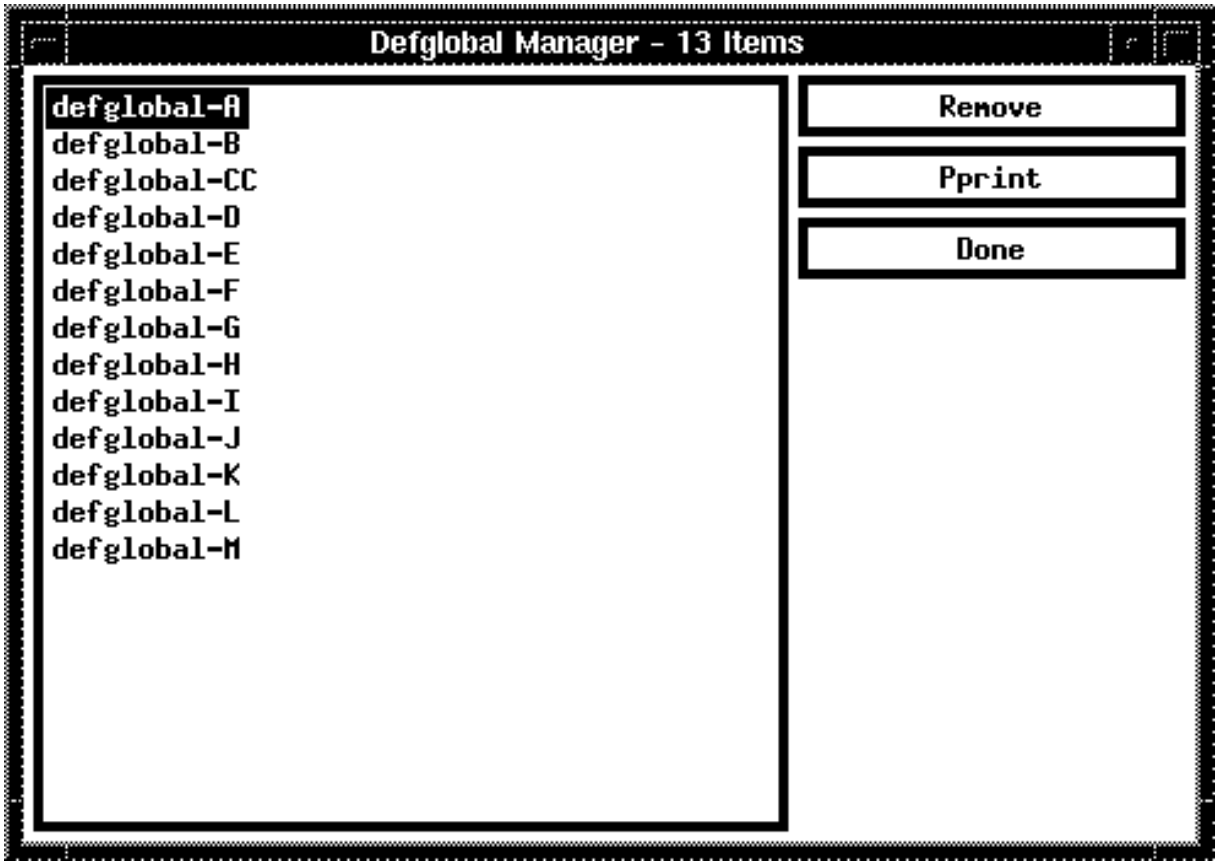
The **Remove** button will remove the currently selected deffunction. Pressing this button is equivalent to an **undeffunction** command. The **Pprint** button will pretty print the currently selected deffunction. Pressing this button is equivalent to a **ppdeffunction** command. The **Pprint** button 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.

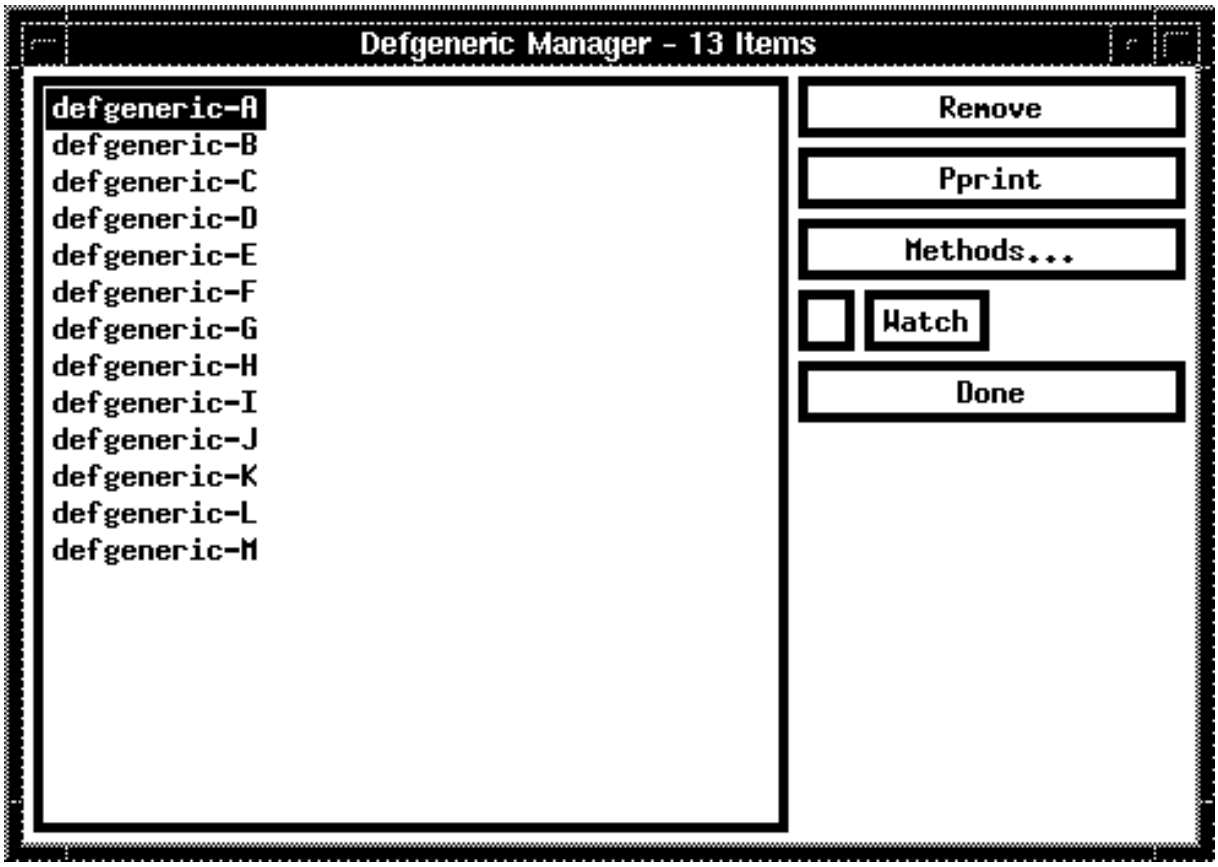


The **Remove** button will remove the currently selected defglobal. Pressing this button is equivalent to an **undefglobal** command The **Pprint** button will pretty print the currently selected deffacts. Pressing this button is equivalent to a **ppdefglobal**

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

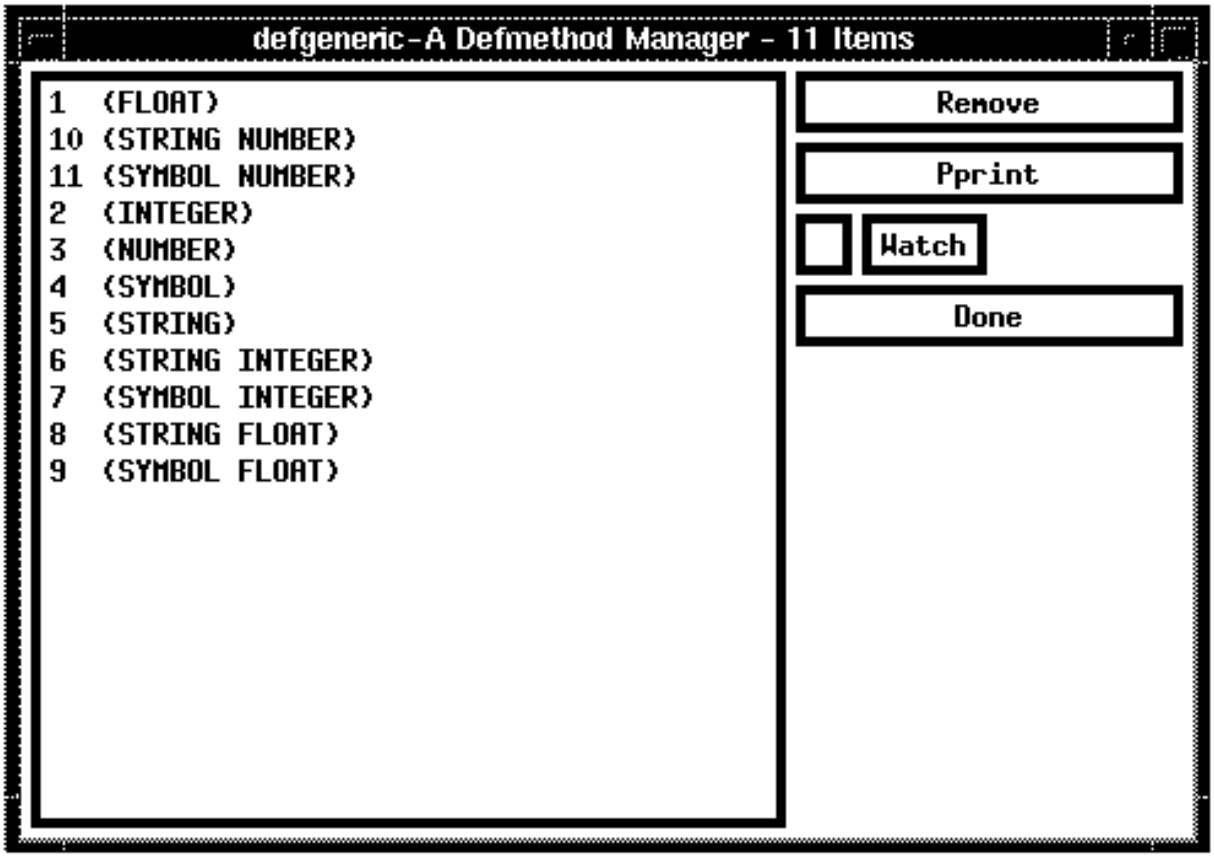
This command displays a dialog box which allows the defgenerics in the knowledge base to be examined. The dialog box that appears in response to this command allows an operation to be performed on a defgeneric selected from the list of defgenerics currently in the knowledge base. Any number of operations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



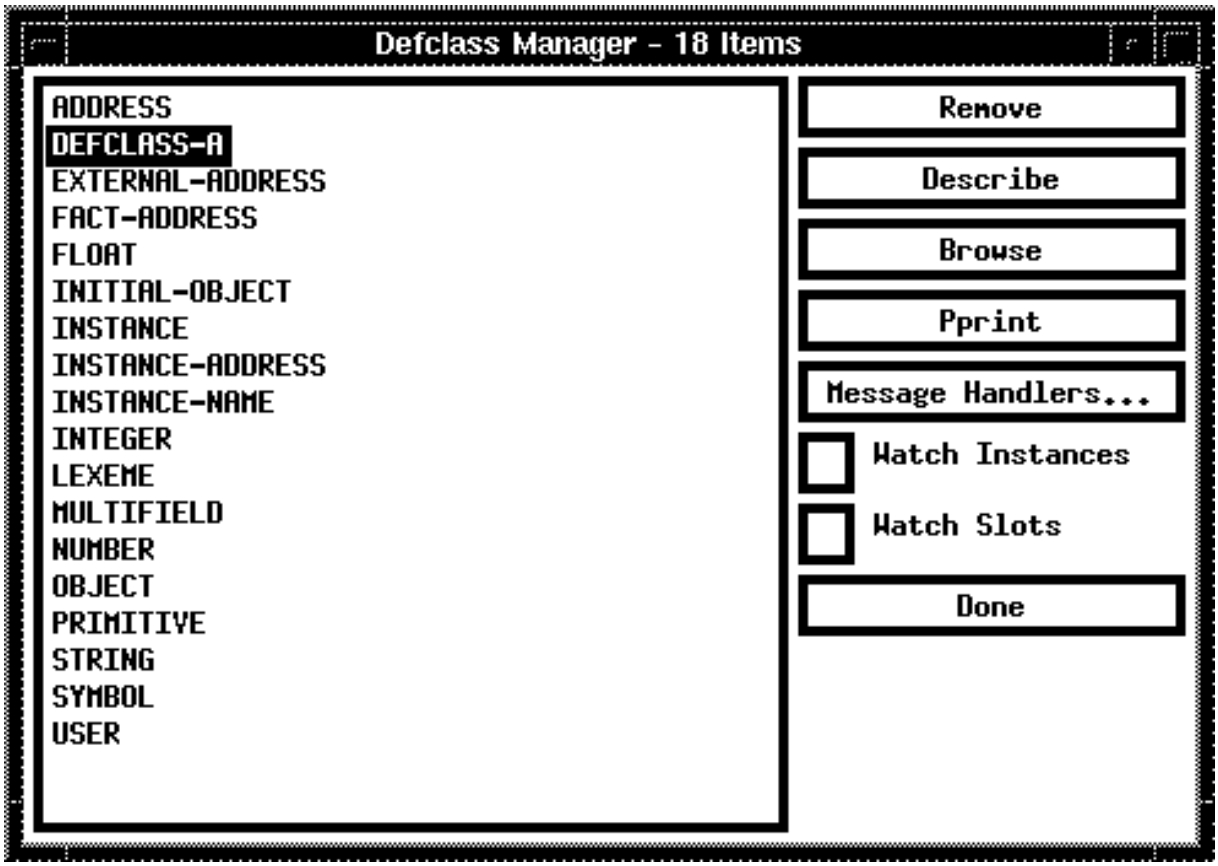
The **Remove** button will remove the currently selected defmethod. Pressing this button is equivalent to an **undefmethod** command. The **Pprint** button will pretty print the currently selected defmethod. Pressing this button is equivalent to a **ppdefmethod** command. The **Pprint** button echoes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

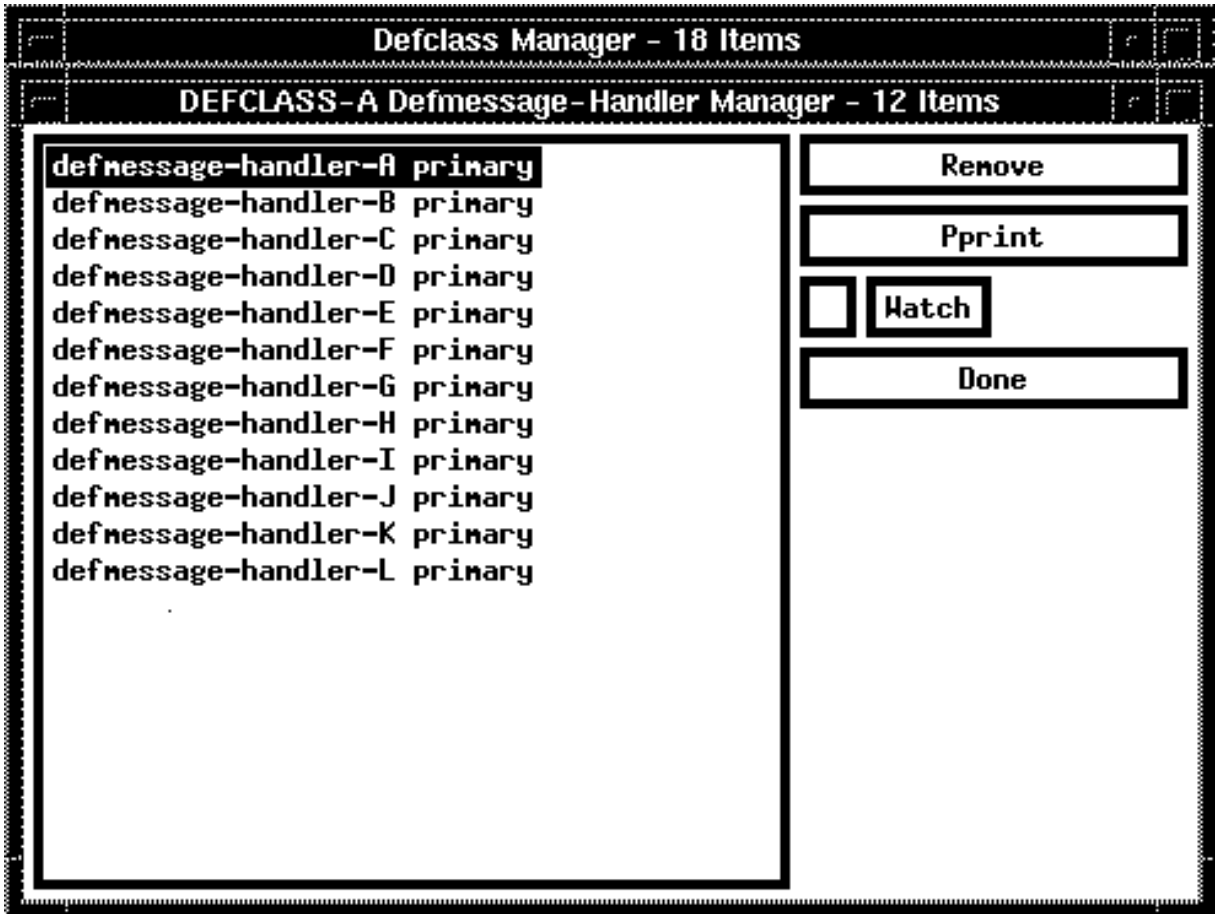


The **Remove** button will remove the currently selected defclass. Pressing this button is equivalent to an **undefclass** command. The **Describe** button will describe the currently selected defclass. Pressing this button is equivalent to a **describe-class** command. The **Browse** command displays 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

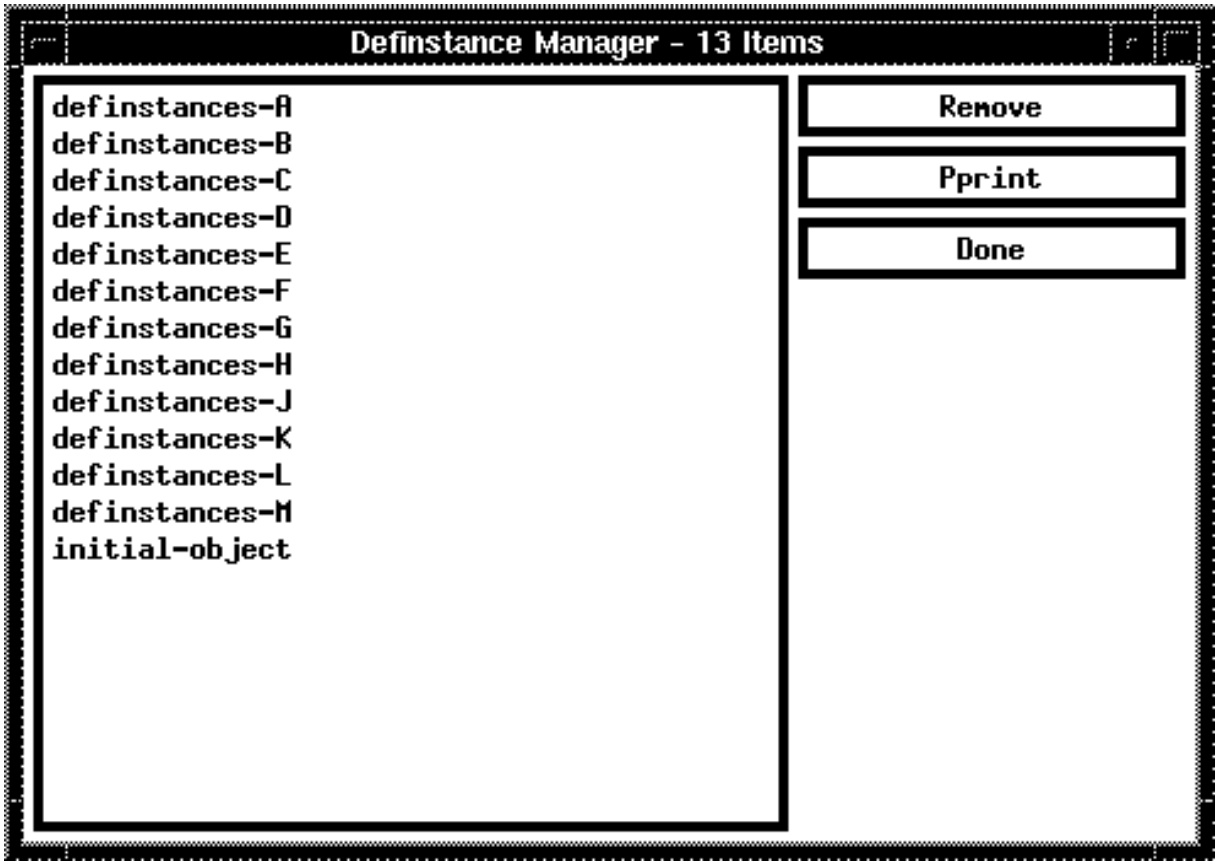
is equivalent to a **ppdefmessage-handler** command. The **Pprint** button echoes the equivalent CLIPS command and resulting output to the dialog window.

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.

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.

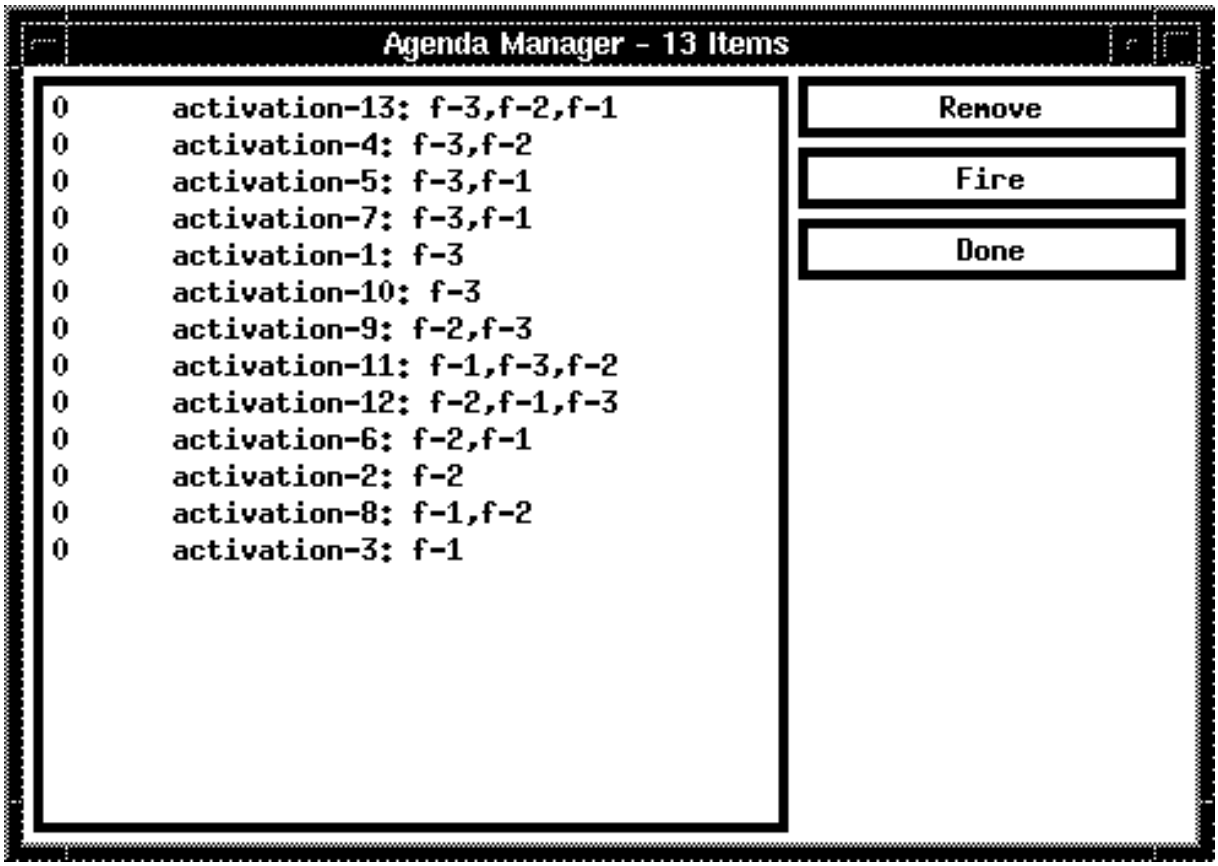


The **Remove** button will remove the currently selected definstances. Pressing this button is equivalent to an **undefinstances** command. The **Pprint** button will pretty print

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.

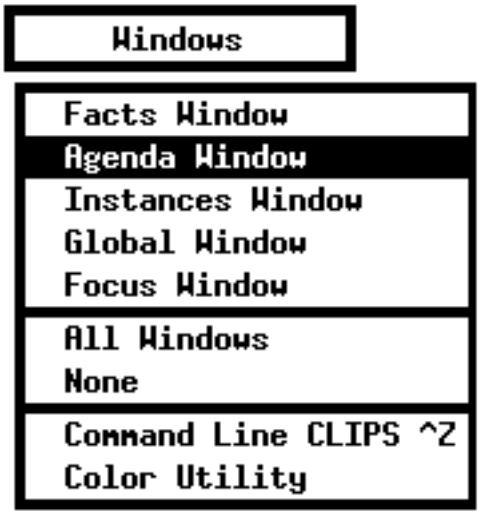
4.4.10 Agenda Manager...

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.



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



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.

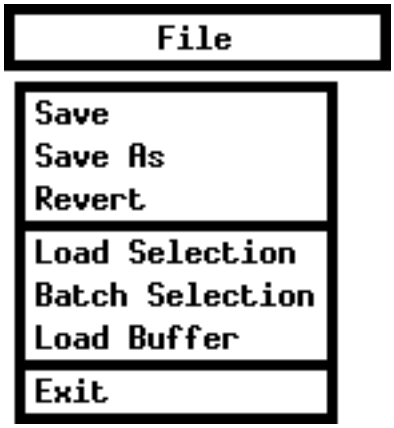
On the top of the color utility there are five buttons the user can select, **Foreground, Background, Border, Default, Quit.** When the foreground and border colors are not allowed to change, the label of the **Foreground** and **Border** buttons will change

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



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.

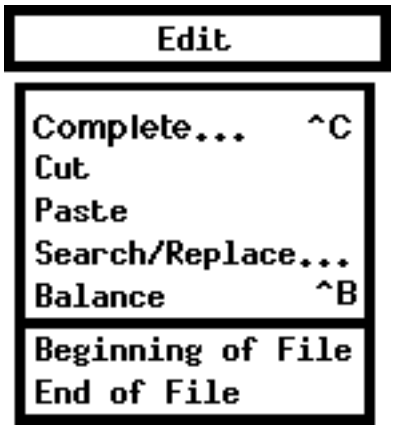
4.6.1.6 Load Buffer

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

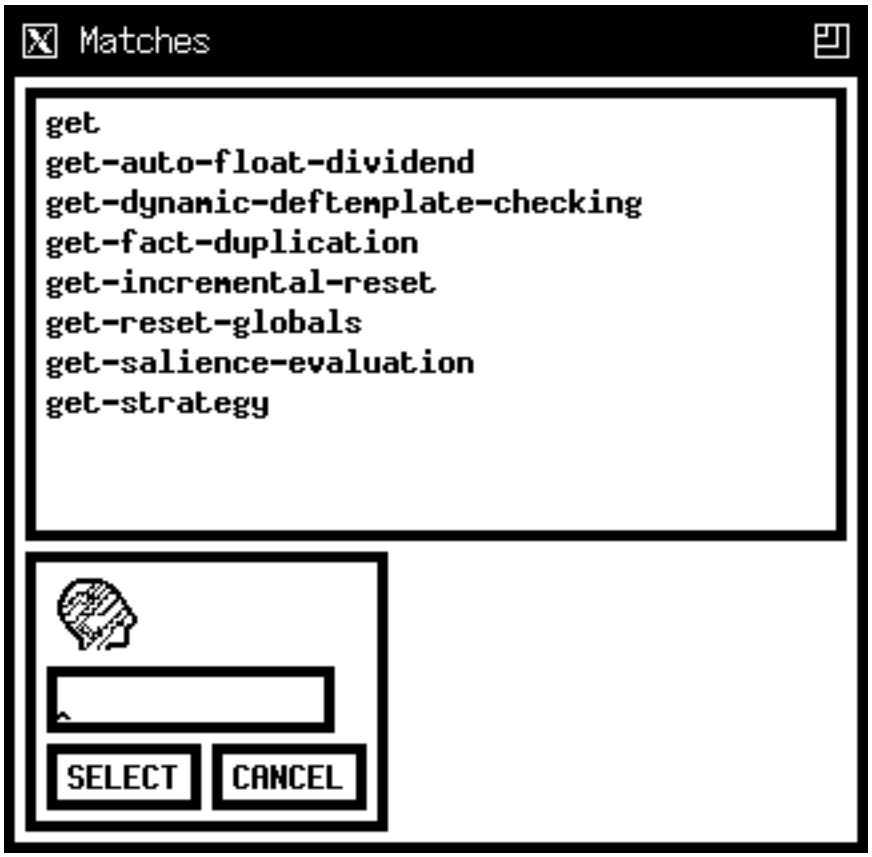
This command exits the editor. This command is not available when CLIPS is executing.

4.6.2 Edit



4.6.2.3 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.

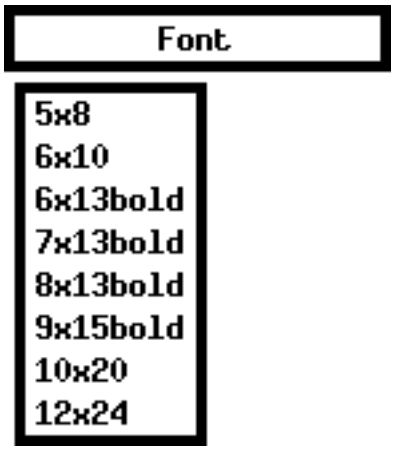
4.6.2.6 Beginning of File

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



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.

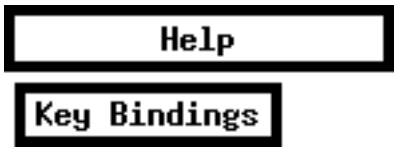
4.6.3.7 10x20

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



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 11.4 and has been tested on Sun-3 and Sun-4 Workstations and Sun Sparcstations. 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	xedit.h	xmain.h	xmenu.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. The compiler directive flag CLP_EDIT in the file **setup.h** should be set to 0.

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:

```
cc -o xclips *.c -lXaw -lXmu -lXt -lXext -X11 -lm
```

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:

```
cc -o color color.c -lXaw -lXmu -lXt -lXext -X11
```