

DINFO

DInfo is a system for browsing graph structured documentation. The only graph currently available for browsing is the *Interlisp-D Reference Manual*. Sadly, tools for editing documentation graphs have not yet been thoroughly developed, although hooks are provided.

User Interface

Loading DINFO.DCOM adds DInfo to the background menu. To start DInfo, one should select DInfo from the background menu. Note that the global variable IRM.HOST&DIR must be set correctly before DInfo is started (see Installing DInfo below). DInfo will take a moment to read in the initial graph, and then be available for use. Most interaction with DInfo is through a menu at the top of DInfo's window, which looks like this:

```

Graph!  IRM
Node:  Saving Virtual Memory State
Top!  IRM Top
Parent!  Miscellaneous
Previous!  Idle Mode
Next!  Date And Time Functions
Display:  Graph  Menu  Text  History
Lookup!  *SYS*
Find!  FILE
  
```

Next to Graph!, Node:, Top!, Parent!, Previous!, and Next! are printed the names of the node currently being visited, the name of the top node in the current graph, the name of the node previous to the current node, and the name of the node next to the previous node, respectively. Graph!, Top!, Parent!, Previous!, and Next! are also commands as follows:

Graph! raises a menu with the names of the graph's DInfo knows about (see DInfo Internals, below). Selecting an item from this menu will cause DInfo to switch to that graph.

Top! visits the top node in the current graph.

Parent! visits the parent of the current node.

Previous! visits the node previous to the current node.

Next! visits the node following the current node.

Thus Previous! and Next! provide sequential access to the graph.

The Display commands control what will be displayed when a node is visited:

Graph toggles display of a Grapher display of the graph local to the current node. Selecting a node in this display will visit the corresponding node in the current DInfo graph.

Menu toggles display of a menu of subnodes of the current node. If the current node has no subnodes no menu will be displayed. Selecting an item in the subnode Menu will visit that node in the graph.

Text toggles display of the text of the current node. Turning this off will speed up the visiting of nodes considerably, useful when searching for a particular node.

History toggles recording and display of a menu containing the history of nodes visited. Selecting an item from this menu will revisit that node.

Lookup! and Find! have editable items to their right in the menu. These items are initially empty, and hence invisible.

Lookup! will look up the term to its right in the index of the *Interlisp-D Reference Manual*, and then bring you to the node containing this term. Note that once you have started DInfo, all manual lookup is modified to work this way, i.e., the ?<CR> TTYIN macro and the HELP lispxmacro will use this facility. Note also that "*" is a wild card in lookup strings.

Find! will search for the string to its right in the text of the current node.

User Variables

DINFOGRAPHWINDOWPOSITION [Variable]

Determines where DInfo will place the window used for displaying the graph. Default is (1 . 1).

DINFOMODES [Variable]

Determines which of the toggles will be selected when DInfo is initially started; it should be a list with recognized members being GRAPH, MENU, TEXT, and HISTORY. Default is (GRAPH TEXT).

DINFO.HISTORY.LENGTH [Variable]

Determines the maximum length of DInfo's history. Default is 10.

Because DInfo sits atop the HelpSys manual lookup facility, the two packages share some resources, including the window. Thus the position of DInfo's window is determined by a variable in HelpSys, namely IRMWINDOWREGION.

Installing DInfo

As DInfo is only currently implemented for the *Interlisp-D Reference Manual*, installation of DInfo is identical to installation of HelpSys. That is, the entire contents of the *Interlisp-D Reference Manual* floppies must be put on one directory (either on a file server or on disk), and the variable IRM.HOST&DIR should be set to the name of this directory.

DInfo Internals

The following information is included for the programmer interested in adding alternate graphs to DInfo.

DINFONODE

[Record]

Contains the following fields:

ID

Unique identifier for node in graph, ala GRAPHNODE field NODEID. Note that EQ is used for checking identity of nodes.

LABEL

The print name of a node. Analogous to the GRAPHNODE field NODELABEL.

FILE

The file containing the documentation for this node. Should not generally include HOST and DIRECTORY fields as DInfo will default these (assuming all documentation files are on one directory, see below).

FROMBYTE

Byte number in FILE where the documentation for this node begins.

TOBYTE

Byte number in FILE where the documentation for this node ends.

DInfo uses OPENTEXTSTREAM to display its files, and thus any TEdit file can be included. Note that if a file has any formatting (image objects in particular), the byte number of a character in a file is not necessarily the same as the TEdit character number of that character. Thus this random access feature is really only useful with plain text files.

PARENT

The ID field of the node parent to this node.

CHILDREN

A list of the ID's of the subnodes of this node.

NEXTNODE

The ID of the next node in the graph.

PREVIOUSNODE

The ID of the node previous to this node in the graph.

USERDATA

Unused. Note that there is no special access function for this field as, for example, WINDOWPROP is for the USERDATA field of a WINDOW. This field is left open for use by implementors for whatever they see fit.

DINFOGRAPH

[Record]

Contains the following fields:

NAME

The name of the graph. Should be an atom. Note that when DInfo reads a graph from a file (with DINFO.READ.GRAPH) this field is set to the NAME field of the file name the graph is read from.

NODELST

The list of nodes in the graph. Each node should be a DINFONODE record.

TOPNODEID

The ID field of the root, or top node of the graph.

CURRENTNODE

Used by DInfo to keep track of where in the graph DInfo is.

DEFAULTHOST

Used if no host is specified in the FILE field of a node.

DEFAULTDIR

Used if no directory is specified in the FILE field of a node.

Note that DEFAULTHOST and DEFAULTDIR are set when a DINFOGRAPH is read from a file (by DINFO.READ.GRAPH) to the host and directory of that file.

TEXTPROPS

Will be passed as the PROPS argument to OPENTEXTSTREAM when the file for a node in the graph is displayed. This feature is used to fake formatting of the *Interlisp-D Reference Manual* files, which are really plain text.

LOOKUPFN

Will be called when the user selects Lookup! from DInfo's FreeMenu with two arguments: The string to look up, and the current DInfo graph.

EDITFN

Called when the middle mouse is depressed in DInfo's window. If not specified, DINFO.DEFAULT.EDITFN will be used. Passed one argument of the current DInfo graph.

USERDATA

Spare field for use by applications.

DINFO.INIT.FILE.NAME

[Variable]

If NIL, DInfo will assume the only graph is in *Interlisp-D Reference Manual*. DINFOGRAPH on *Interlisp-D Reference Manual*. HOST&DIR, otherwise DInfo will READFILE the file named by this variable to find out what graphs it knows about. READFILE of this file should return a list, the CAR of which is discarded (assumed to be a comment) and the CDR of which should be a list of files containing DINFOGRAPHS (written by DINFO.WRITE.GRAPH, of course). You should have gotten a sample DINFO.INIT somewhere with this release.

DINFOW

[Variable]

Bound to the window that DInfo uses. One should never set this variable, as DInfo uses it to determine whether DInfo has been initialized. WINDOWPROPS of interest:

CURRENT.GRAPH

[Window property]

The current DINFOGRAPH.

GRAPH.FILES

[Window property]

A list of file names of graphs DInfo knows about. Used to create the menu raised by the Graph! command in DInfo's FreeMenu.

DInfo caches lots of things as WINDOPROPS of DINFOW. A wise programmer would do well to look here if in search of some obscure aspect of DInfo's state.

(DINFO.UPDATE *NODE SEL*)

[Function]

Will visit node in current DInfo graph. *NODE* should be a DINFONODE record. *SEL* is used by DInfo's *Interlisp-D Reference Manual* lookup facility, and should be useful in implementing other lookup facilities. *SEL* determines what in the TEXT of this node will be selected. *SEL* should be a list of the format (*NAME X . Y*) where *NAME* is the name of the selection, *X* is the character number in the text of *NODE* where *SEL* starts, and *Y* is the length of the selection. If *Y* is NIL (i.e., *SEL* = (*NAME X*)) then the length of name is used as the length of the selection. This is useful for the lookup of terms.

(DINFO.RESET)

[Function]

Will completely reset DInfo.

Hopefully there will be a full editing interface in the next release. This would allow the user to add and and change graphs, allowing DInfo to act as a document organizer.

HELPSYS

HelpSys allows on-line lookup and display of references from the *Interlisp-D Reference Manual*. HelpSys enables the TTYIN macro `?cr`, which works much like the TTYIN macro `? = cr`. The `?cr` macro prints the manual entry for the function in question, whereas `? = cr` just displays the argument list. One can also access this facility through the HELP LISPXMACRO (try HELP <keyword> to the Interlisp Executive) and by using the Lookup button atop HelpSys's window. Note that starting the DInfo system (another Library package) significantly modifies how a manual reference is displayed (for the better, of course).

(IRM.LOOKUP KEYWORD TYPE WINDOW) [Function]

This is the primary function of HelpSys. If *TYPE* is specified, then the primary manual entry for *KEYWORD* of *TYPE*, or the first if there is no primary, will be displayed in *WINDOW*. If *TYPE* is not specified, then a pop-up menu will be raised containing all the manual entries for *KEYWORD*. Primary entries are marked with stars (*'s) on either end of the item. *WINDOW* defaults to IRMWINDOW.

(IRM.SMART.LOOKUP KEYWORD WINDOW) [Function]

Uses wild card matching if *'s are in *KEYWORD* (* matches any substring) or tries spelling correction. A pop-up menu is raised if more than one wild card match is found. Note that the first time a * appears in a *KEYWORD*, HelpSys will need to load the list of possible keywords for matching against, and only after this list has been loaded will spelling correction be enabled. This is the function called by the HELP LISPXMACRO, by the Lookup button atop IRMWINDOW, and by DInfo's *Interlisp-D Reference Manual* Lookup facility.

(HELPSYS FN —) [Function]

This is the function called by TTYIN when you type `?cr` with the CAR of the current form being typed as *FN*. It figures out what type *FN* is (Function, Macro, or I.S.Operator etc. . .) and calls IRM.LOOKUP appropriately.

User Variables

IRMWINDOW [Variable]

This is the default window used by HelpSys for printing manual references.

IRMWINDOWREGION [Variable]

HelpSys uses this when creating IRMWINDOW. Note that if IRMWINDOW has been opened you must (SETQ IRMWINDOW NIL) for this variable to take effect.

IRM.FONT [Variable]

This should be set to a display font descriptor and determines the font that will be used for the printing of manual entries. Initially set to the value of:

(FONTCREATE 'HELVETICA 10 NIL NIL 'DISPLAY).

Installing Helpsys

Helpsys requires the text files of the *Interlisp-D Reference Manual* to be on a random access filing device, i.e., either {DSK} or a file server. The files it requires are Chap*.IRM and IRM.HASHFILE on the Lisp Library floppies. To install HELPSYS you must copy all these files to one directory, and set the variable IRM.HOST&DIR to the name of this directory.

IRM.HOST&DIR [Variable]

This determines where HelpSys will look for the manual text files and hash file. This should be set in your site init file. As file servers can often be quite loaded down, HelpSys will work much faster if you move these files to {DSK}.

IRM.HASHFILE.NAME [Variable]

If this is NIL, HelpSys will look for the hash file as IRM.HASHFILE on IRM.HOST&DIR, otherwise it uses the value of this variable. To speed up HelpSys one can copy the hash file to {DSK}, set this variable to its new location, and set \IRM.HASHFILE to NIL so that HelpSys will be forced to use the new hash file.