## A User's Guide to TEdit—Release Notes

For the Medley Release, TEdit has increased the number of expanded characters, added options to the **Put** and **Get** submenus, clarified several options in the Paragraph Looks and Page Layout menus, and added several minor items to the programmer's interface.
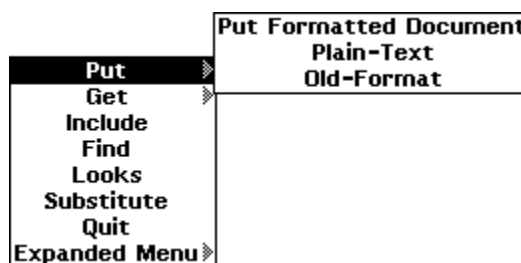
### Expanded Characters

TEdit added the following abbreviations and expansions to the characters shown in Table 1.

*Table 1.   TEdit's Abbreviations and their Expanded Characters*

| Abbreviation | Expanded character name | Expansion Character |
|:---:|:---|:---:|
| p | Pilcrow (proofreader's paragraph mark) | ¶ |
| t | Trademark | ™ |
| tm | Trademark | ™ |
| r | Registered trademark | ® |
| 1/3 | Built-up fraction | ⅓ |
| x | Times sign | × |
| / | Division sign | ÷ |
| o (oh) | Degrees sign | ° |
| L | Pound sterling sign | £ |
| Y | Yen sign | ¥ |
| + | Plus-or-minus sign | ± |
| ^(shift-6) | Up arrow (NS character) | ↑ |
| ua | Up arrow (NS character) | ↑ |
| &#124; | Down arrow (NS character) | ↓ |
| da | Down arrow (NS character) | ↓ |
| <- | Left arrow (NS character) | ← |
| la | Left arrow (NS character) | ← |
| _ (underscore) | Left arrow (NS character) | ← |
| -> | Right arrow (NS character) | → |
| ra | Right arrow (NS character) | → |
| = | Two-way arrows (NS characters) | ⇔ |

### Put Submenu

The drag-through menu for the **Put** command now has the following entries:

```
                              Put Formatted Document
                                  Plain-Text
                   Put      »     Old-Format
                   Get      »
                Include
                   Find
                  Looks
              Substitute
                   Quit
          Expanded Menu »
```
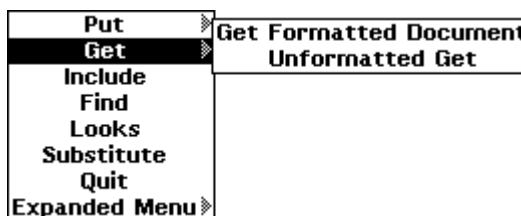
The **Put** command has a submenu that offers you several options for saving your file:

• Keep the formatting in the file. Use this **Put Formatted Document** option, which is the default, unless you have special requirements.

• Save the file as plain text, regardless of formatting. Using this **Plain-Text** option removes all of TEdit's formatting from the file, leaving plain text.

• Save TEdit files in an "old" format. This **Old-Format** option allows you save files in the format of a previous release of TEdit. This format option is provided for backward compatibility.

## Get Submenu

The drag-through menu for the **Get** command now has the following entries:

```
                   Put      » Get Formatted Document
                   Get      »    Unformatted Get
                Include
                   Find
                  Looks
              Substitute
                   Quit
          Expanded Menu »
```

**Get** has a submenu that offers you the option of retrieving a formatted file (**Get Formatted Document**), or retrieving a file as though it were plain text, with most formatting information appearing as black rectangles ( • ).

# Clarified Paragraph Looks Menu Options

Both the menu options **New Page: Before After** and **Displaymode: Hardcopy** have expanded explanations.

## New Page: Before After

Sometimes a page break occurs so that the first paragraph on a page is marked with the **Before** command. In these cases, the text flows continuously from the previous page to this page; a blank page does not appear between them.

**Displaymode: Hardcopy**

The Hardcopy displaymode command now works only when the text is printed in Interpress fonts.

## Clarified Page Layout Menu Options

When specifying text to appear before or after page numbers, you can only enter text in the brackets; image objects are not allowed.

You may now specify a landscape page layout.

In the page layout menu, Modern 10 MRR is now the default page number font instead of Gacha 10. Also, there is a global variable, TEDIT.DEFAULT.FOLIO.LOOKS, that you can set to be any character-looks specification acceptable to TEDIT.LOOKS. The default (i.e., if you don't specify one in the page layout menu) is taken from there.

If you have set page formatting in the past, the page-numbering font has been set as well (even if you specified nothing). This behavior continues, but the default is more sensible, and can be changed.

You may now number the first page of a TEdit file 0 (zero).

TEdit now preserves text before and after page numbers after a file is saved.

Using numbers with decimal points in the "Text before page number" field in the page-layout menu now works properly.

## Added Items to Programmer's Interface

The TEXTOBJ data structure has a correction, the TEDIT.INCLUDE, TEDIT.PARALOOKS and TEXTPROP functions are expanded, and the global variable TEDIT.KNOWN.FONTS is now documented.

### Corrected the AFTERQUITFN Property

(AFTERQUITFN *WINDOW TEXTSTREAM*) is an optional user-supplied Lisp function that is called after ending an editing session to peform any required cleanup. The *WINDOW* argument was omitted in the manual.

## Corrected the TITLEMENUFN Property

TEDIT.TITLEMENUFN is a window property, not a TEdit property as documented in the manual.

## Corrected the TEXTOBJ Data Structure

The data structure called TEXTOBJ has as its first field of interest \WINDOW, not WINDOW as documented in the manual.

## Expanded the TEDIT.INCLUDE Function

TEDIT.INCLUDE now accepts optional START and END arguments that instruct it to restrict its attention to a portion of the TEdit file, treating this part as a separate file. This feature is useful when you require that several distinct TEdit documents reside within a single TEdit file, for example, for database applications. Each document can be formatted and extracted separately.

---

**CAUTION**

If you use START and END arguments with INCLUDE, and then format the entire TEdit file, you will lose the formatting.

---

**CAUTION**

TEDIT.INCLUDE and OPENTEXTSTREAM take optional arguments that let you take a document out of the middle of a file. This option requires that Lisp be able to determine the length of the file before it is read. Some file protocols (TCP FTP in particular) don't let Lisp do this. If you try to use this option with a file that resides at the other end of a TCP connection (or, more generally, on any device where you cannot tell the length of the file until you have read the whole file), it won't work. The result will be that your document will contain no characters.

---

## Expanded the TEDIT.PARALOOKS Function

TEDIT.PARALOOKS can be used to set NEWPAGEBEFORE, NEWPAGEAFTER, HARDCOPY, TYPE, SUBTYPE, REVISED, KEEP, STYLE, CHARSTYLES, and USERINFO parameters.

REVISED, if non-NIL, causes a revision bar to be printed one pica to the right of the right margin of the paragraph. It is a vertical bar 1 point wide from the top of the top line's ascent to the bottom of the bottom line's descent.

USERINFO can be used as a property list for saving information of interest to the user. It is generally used in a number of undocumented features (e.g. footnote support).

KEEP, STYLE, and CHARSTYLES are reserved for a future release.

## Expanded the TEXTPROP Function

You can now also use your own properties, but these properties are not saved with the document if you **Put** it.

## Added Documentation for a Global Variable

The following documentation should be added to TEdit's Global Variables.

TEDIT.KNOWN.FONTS                                                   [Variable]

A list of available fonts that appear in the Character Looks menu. The list in in the form ((name-in-the-menu-1 'Real-font-name-1) (name-in-the-menu-2 'Real-font-name-2) ...), for example, ((Classic 'CLASSIC) (Times 'TIMESROMAN)).

# Changes to Programmer's Interface to TEdit

## STREAM AND TEXTOBJ

All public TEdit functions (non- \\) that take a *TEXTOBJ* argument accept either a TEXTOBJ or a text STREAM as that argument's value.

## Changes, Additions and Corrections to TEdit functions

The function TEDIT.SINGLE.PAGEFORMAT is incorrectly documented in the Lisp Library. The following corrections should be noted: The arguments *PG#X*, *PG#Y*, and *PG#FONT* should be *PX, PY*, and *PFONT*, respectively.

The argument *PG#ALIGNMENT* should be *PQUAD*.

The order for the arguments, *TOP BOTTOM LEFT RIGHT* should be *LEFT RIGHT TOP BOTTOM*.

The argument *#COLS* should be *COLS*.

*INTERCOLSPACE* should be *INTERCOL*. And between the *INTERCOL* and *UNITS* arguments there is a *HEADINGS* argument.

The functions and its arguments look like:

(TEDIT.SINGLE.PAGEFORMAT     *PAGE#S? PX PY PFONT PQUAD LEFT RIGHT*
                   *TOP BOTTOM COLS COLWIDTH INTERCOL*
                   *HEADINGS  UNITS PAGEPROPS PAPERSIZE*)      [Function
]

*PAGE#S?*    T if you want page numbers on this kind of page, else NIL.

*PX*    The horizontal location of the page number, measured from the left edge of the paper. Negative values are measured from the paper's right edge.

| | |
|---|---|
| *PY* | The vertical location of the baseline for the page numbers, measured from the bottom of the paper. Negative values are measured from the top of the paper. |
| *PFONT* | The font to be used to display the page numbers. This can be any specification that is acceptable to TEDIT.LOOKS. |
| *PQUAD* | An atom that tells how the page number is to be aligned on the location specified by *PX* and *PY.* LEFT means the location is the lower-left corner of the page number. RIGHT means the location is the lower-right corner. CENTERED means the page number will be centered around the *PX* you specified. |
| *LEFT* | The left margin—the distance from the left edge of the paper to the left edge of the first text column. |
| *RIGHT* | The right margin—the distance from the right edge of the rightmost text column to the right edge of the paper. |
| *TOP* | The top margin of the page—the distance from the top of the paper to the top of the first line of body text. |
| *BOTTOM* | The bottom margin—the distance from the bottom of the last line of body text to the bottom of the paper. |
| *COLS* | Number of columns (default is one). |
| *COLWIDTH* | The column width (default is to evenly divide the available space among the *#COLS* columns). |
| *INTERCOL* | The space between the right edge of one column and the left edge of the next column. Defaults to evenly divide the space left after the columns are set up. If there is more than one column, one or the other of *COLWIDTH* and *INTERCOLSPACE must* be specified. |
| *HEADINGS* | A list of lists in the form of ((HEADINGNAME$_1$ XLOCATION$_1$ YLOCATION$_1$) (HEADINGNAME$_2$ XLOCATION$_2$ YLOCATION$_2$) . . .(HEADINGNAME$_n$ XLOCATION$_n$ YLOCATION$_n$)). |
| *UNITS* | The units used in setting the values you specified. May be one of the atoms PICAS, IN, INCHES, CM, POINTS. Default is POINTS. |
| *PAGEPROPS* | A property list of extra information. Properties are STARTINGPAGE#, FOLIOINFO, and LANDSCAPE?. |
| | STARTINGPAGE# is the first page's number; it is ignored if this isn't the first page. |
| | FOLIOINFO is a list of information about page numbers, (FORMAT TEXTBEFORE TEXTAFTER). FORMAT can be one of ARABIC, LOWERROMAN, UPPERROMAN, or NIL (i.e., ARABIC). TEXTBEFORE is the text preceding the number, and TEXTAFTER is the text following the number. |
| | LANDSCAPE? determines if the document is printed in the usual vertical format or printed in landscape format (horizontally). If NIL the document is printed vertically, if non-NIL the document is printed landscape. Defaults to NIL. |
| *PAPERSIZE* | Is one of LETTER, LEGAL, the metric paper sizes (A0, A1, A2 A3, A4, A5, B0, B2, B3, B4), or NIL (which defaults to letter size). |

TEDIT.GET accepts an open stream as the file to GET from.  You may still pass it a TEXTOBJ, however.

(TEDIT.GET *STREAM FILE UNFORMATTED*?)                    [Function]

Performs the TEdit Get command, loading the text from *FILE* onto the editing stream *STREAM*—replacing the text that is being edited currently.  If *FILE* is not supplied, the user will be asked for a file name.  If *UNFORMATTED?* is non-NIL, *FILE* is treated as a plain-text document, and all of its contents are included—even TEdit formatting information.

You can now use TEDIT.PUT to store a TEdit document in the middle of a larger file (e.g., for saving TEdit documents as part of a database).  The complete documentation is now as follows:

(TEDIT.PUT *STREAM FILE FORCENEW  UNFORMATTED? OLDFORMAT*?) [Function]

Performs the TEdit Put command, saving the text from the text stream *STREAM* onto the file named *FILE.*  If *FILE* is NIL, the user will be prompted for a file name.  In this case, if *FORCENEW* is NIL, the user is offered the old file name as a default; if non-NIL, no default is given, forcing the user to specify a file name.  If *UNFORMATTED?* is non-NIL, only characters are put in the file—no formatting.  If *OLDFORMAT?* is non-NIL, the file will be written in the format used by the previous version of TEdit, for backward compatibility.

In order to store a TEdit document as part of another file, call TEDIT.PUT, passing an open stream on the file as the *FILE* argument.  The stream should be open for output and positioned at the place you want TEdit to store the document (call this file pointer START).  When TEDIT.PUT returns, the stream's end-of-file pointer will be just after the last byte in the newly-inserted document.  Call this file pointer END.  To subsequently retrieve the document from the middle of this other file, call OPENTEXTSTREAM on the file, passing the START and END pointers as the *START* and *END* arguments.

Note:                When TEDIT.PUT returns, the stream will be open for INPUT.

The functions TEDIT.MOVE and TEDIT.COPY were not documented in Koto.  They are:

(TEDIT.MOVE *FROM TO*)                                  [Function]

*FROM*  and *TO* are SELECTIONs.  Moves the text described by *FROM* to the place described by *TO*, within the same text stream or between different text  streams.  The text described by *FROM* is deleted from its original location.

(TEDIT.COPY *FROM TO*)                                  [Function]

*FROM*  and *TO* are SELECTIONs.  Copies the text described by *FROM* to the place described by *TO*, within the same text stream or between different text  streams. The text described by *FROM* is not deleted in the *FROM*  location.

## Changes in the Documentation of TEdit Functions

The following functions have had the documentation of their arguments changed to reflect what will appear if you do a ?= or evaluate ARGLIST on one of these functions. Arguments that were corrected are indicated by bold italics (***arg***). Please note that what changed was the documentation, not the way the functions operate or the values of the arguments themselves.

(TEDIT.SETSEL                                                  *STREAM **CH#** LEN POINT **PENDINGDELFLG***
        *LEAVECARETLOOKS OPERATION*)                      [Function]

(COERCETEXTOBJ *STREAM TYPE **OUTPUTSTREAM***)          [Function]

(TEDIT.DELETE *STREAM **SEL** LEN*)            [Function]

(TEDIT.INCLUDE ***STREAM** FILE START END*)        [Function]

(TEDIT.FIND ***STREAM TARGETSTRING** START# END#  WILDCARDS?*)  [Function]

(TEDIT.GET.LOOKS ***STREAM CH#ORCHARLOOKS***)        [Function]

(TEDIT.PARALOOKS ***STREAM** NEWLOOKS **SEL LEN***)       [Function]

(TEDIT.COMPOUND.PAGEFORMAT ***FIRST VERSO  RECTO***)    [Function]

(TEXTOBJ ***STREAM***)                [Function]

(TEXTSTREAM ***STREAM***)              [Function]

(TEDIT.CARETLOOKS *STREAM **LOOKS***)           [Function]

(TEDIT.NORMALIZECARET ***STREAM** SEL*)         [Function]

(COPYTEXTSTREAM ***ORIGINAL** CROSSCOPY*)         [Function]

(TEDIT.PROMPTPRINT ***TEXTSTREAM** MSG CLEAR?*)      [Function]

(TEDIT.SETSYNTAX ***CHAR** CLASS TABLE*)        [Function]

(TEDIT.GETSYNTAX ***CH** TABLE*)           [Function]

(TEDIT.SETFUNCTION *CHARCODE FN **RTBL***)         [Function]

(TEDIT.WORDGET ***CH** TABLE*)            [Function]

(TEDIT.WORDSET ***CHARCODE** CLASS TABLE*)       [Function]

(TEDIT.INSERT.OBJECT *OBJECT STREAM **CH#***)       [Function]

The following functions were previously documented as accepting a TEXTOBJ. They all still take a TEXTOBJ but they will now also accept a STREAM as the first argument.

(TEDIT.FIND *STREAM TARGETSTRING START# END#  WILDCARDS?*)  [Function]

(TEDIT.GET.LOOKS *STREAM CH#ORCHARLOOKS*)        [Function]

(TEDIT.PARALOOKS *STREAM NEWLOOKS SEL LEN*)       [Function]

(TEXTSTREAM *STREAM*)               [Function]

(TEDIT.NORMALIZECARET *STREAM SEL*)         [Function]

(TEDIT.PROMPTPRINT *TEXTSTREAM MSG CLEAR?*)      [Function]

## New Features

For the benefit of NS file server users, TEdit now writes files of type TEDIT, instead of BINARY. As a result, LISTFILES and the FileBrowser are able to determine that the file is a TEdit file and call

TEdit to create the hardcopy. Previously, it was necessary that the TEdit file explicitly have the extension ".TEdit".

```
(OPENSTREAM file 'OUTPUT 'NEW '((TYPE TEDIT))).
```

This change is for formatted files only. Plain text files are still written as type TEXT. Also, on devices that don't support arbitrary file types (such as conventional mainframe file servers), the type TEDIT coerces to BINARY. Unfortunately, if you subsequently copy the file to an NS file server from such a device, the knowledge of its "true" file type is lost.

## A User's Guide to Sketch—Release Notes

The Medley release of Sketch includes several new features, many added in response to user's requests. A programmer's interface allows sketches to be created by programs. This interface is described in a separate document (*The Programmer's Interface to Sketch*.)

## Manipulating Sketch Elements

### Adding and Deleting Control Points

Individual control points can now be added to and deleted from wires and curves.

### Deleting Control Points

You now have the option to delete elements or delete a control point. Just select the **Delete** command, move the mouse cursor out through the grey arrow, then select the point to be deleted.

### Defaults Command

#### Better Feedback for Creating Wires, Circles and Ellipses

Sketch now provides better feedback when you are creating circles, ellipses and wires. You are now prompted with an image of what the figure will look like if you release the left button. You can get the old feedback behavior (for example, if this is too slow) by selecting the **Feedback** subcommand from the **Defaults** submenu, then selecting the **Points only** subcommand from its submenu.

### Arrowheads

A curved arrowhead shape was added and is now the default. Also, a command was added to the menu of arrowhead change operations that implements "look same" for arrowheads. To make the arrowheads on a collection of elements look the same: select **Change**; then, when prompted to select the elements to change, first select the element that has the desired arrowhead, then, in the same selection, add the elements that you want to look like the first one; then select the item **Arrowheads**, then the item **Both**, then the item **Same as First**.

### Deleting Characters During Type-in

You can now delete characters by using the UNDO key, just as you would in TEdit. Type in a word or a phrase, then press the UNDO key, and the text will be deleted.

## Using Bit Maps in  a Sketch

### Zooming Bitmaps

The bit image element provides a bitmap that zooms.  Selecting the **Bit image** command from the command menu will prompt you for a region of the screen that will be inserted as a bit image into the sketch.

### Changing Bitmaps

When you apply a **Change** command to a bit image that it is being viewed at actual size, you will be prompted with the same menu as a bitmap image object.  If the image is being displayed at other than original scale, you will be given the menu shown below.

```
                       Scaled bitmap operations
Perform edit operations on the source bitmap of this image.
         Make the image shown be the source
         Make the source be at this scale
Make the image shown be the source at the source scale
Save this image to be used as a source at this scale
```

*Menu of commands offered when **Change** command is applied to a bit image that is not at the original scale.*

## Freezing Sketch Elements

It is now possible to freeze elements, that is to make them unaffected by edit changes.  Frozen elements will not have their control points highlighted (and hence cannot be selected) after an edit command has been selected.  This provides a way to keep part of the figure fixed while editing on an overlapping part.  It also reduces the number of control points.  The **Freeze** command is a subcommand to the **Group** command.  It will prompt you for a collection of elements that will then be frozen.  Elements can be unfrozen by the **UnFreeze** command that is a subcommand to the **UnGroup** command.

## Aligning Sketch Elements

Sketch contains a set of commands to align elements.  The main menu command **Align** prompts for a collection of control points and moves them so that they all line up with the leftmost one.

## Placing Multiple Copies of Elements

There is a new feature in Sketch that makes it much easier to place multiple copies of a collection of elements.  While positioning the image of the elements during the **Copy** command, hold down the COPY key.   A new copy of the elements will be positioned everytime a mouse button (left or right) is pressed and released, until either the image is placed completely outside the viewer or the COPY key is released before the mouse button is released.

## Making  the Window Fit the Sketch

The **Fit to window** subcommand under the **Move View** command will zoom the sketch so that it just fits within the current window.  It

has a sub-subcommand **Fit window to sketch** that will reshape the window so that the entire sketch (at the size shown) just fits within it. This is useful if you change a sketch that was edited from a document.

## Overlaying Figure Elements

Elements that have a filling property (boxes, text boxes, circles, polygons and closed curves) now have a mode property that determines how the filling should effect elements it covers. The option **Filling mode** now appears in the **Which aspect?** submenu.

## Changing How Elements Overlap

Elements have an order in which they are displayed. An element that is displayed early can be covered by elements layed down later. Thus, changing the order in which overlapping elements are displayed can effect the resulting image. The **Bury** command provides three subcommands to change the order in which elements are displayed.

The **Bury** command will prompt you to select an element or elements and will change their order so that they are displayed first. That is, they will appear underneath any other elements. If you select more than one element, they will all be displayed before any non-selected elements and their relative order maintained. The **Send to bottom** subcommand does the same thing as **Bury**.

The **Bring to top** command is a subitem to the **Bury** command. It will prompt you to select an element or elements and will change their order so that they are displayed last. That is, they will appear on top of any other elements. If you select more than one element, they will all be displayed after any non-selected elements and their relative order maintained.

The **Reverse order** command is a subitem to the **Bury** command. It will prompt you to select a collection of elements and will reverse their display orders. A special case is when two elements are selected. In this case the element positions are switched.

## Loading the Sketch Library Module in the 1186 Environment

The SKETCH executable files are too large to be contained on one floppy. The files are now distributed on two floppies: *Medley Library Floppy #3* and *Medley Library Floppy #4*. To load SKETCH, type the Interlisp exec command:

(FILESLOAD LOAD-SKETCH)

The LOAD-SKETCH function will copy all SKETCH files from #3; then prompt you to insert #4, and the remainder of the files will be copied.

# The Programmer's Interface

The programmer's interface allows Sketch to be used as a tool by other programs. It is documented in t*he Programmer's Interface to Sketch* .

## New Behavior for the Get Command

The action of the **Get** command was changed to be consistent with the TEdit **Get** command. It now deletes any sketch elements that are in the sketch prior to the **Get** command. The affect of the old **Get** command is available as the **Include** command on a submenu to the **Get** command.

## Establishing Initial Defaults for Sketch

The variable SK.DEFAULT.FONT, if non-NIL, is used as the default font. If SK.DEFAULT.FONT is NIL, the default font (DEFAULTFONT) is used.

The following variables are used to establish the default setting for a new sketch. Descriptions of legal values can be found in the *Programmer's Interface to Sketch*. SK.DEFAULT.BRUSH is the default brush. SK.DEFAULT.ARROW.LENGTH is the default arrowhead size. SK.DEFAULT.ARROW.TYPE is the default type (one of LINE, CURVE, CLOSEDLINE or SOLID). SK.DEFAULT.ARROW.ANGLE is the default angle for arrowheads. SK.DEFAULT.TEXT.ALIGNMENT is the default text alignment. SK.DEFAULT.TEXTBOX.ALIGNMENT is the default textbox alignment. SK.DEFAULT.DASHING is the default dashing. SK.DEFAULT.TEXTURE, SK.DEFAULT.BACKCOLOR and SK.DEFAULT.OPERATION are combined to create the default filling.

# 1108 User's Guide Release Notes

## What to Look For

The *1108 User's Guide* was extensively reorganized and rewritten for the Lyric Release. This made it nearly identical to the *1186 User's Guide*. This section contains a summary of changes affecting 1108 environments with the Medley release. Details are described in update pages available for the *1108 User's Guide.*

In every 1108 chapter that requires use of Lisp expressions of any kind, there is a notice regarding the use of **IL:** and a suggestion that expressions, functions, and variables be typed into an Interlisp Exec.

## 4. File System

Medley will accept floppy names up to 40 characters in length. Some of the Lyric font floppies have names in excess of 40 characters. Medley truncates the floppy name to 40 characters if asked to read a Lyric floppy with a longer name. The function FLOPPY.NAME is used to name a floppy. When it is not given any arguments, it returns the name stored on the floppy disk. When it is given a *NAME* argument, the floppy name is set to *NAME*. The 40 character limitation holds for both 1108 and 1186 floppies.

The {DSK} device on the 1108 and 1186 now accepts a wider range of characters in file names. Almost any character in character set 0 is acceptable. Previously, if you tried to create a file whose name included, for example, an underscore, you would see a "FILE NOT FOUND" error.

The 1108 and 1186 file systems had a problem with large partitions which would manifest itself as "HARD DISK error - can't find file page" when accessing newly created files. This would only appear on logical volumes larger than 64K pages. This problem has been fixed.

The function FILENAMEFROMID is now implemented.

## 6. System Tools

In System Tools, it is no longer necessary to execute a **Floppy Info!** command before attempting a **List!**.

The Medley System Tool now displays an error message when an NS Domain or Organization name is more than the allowed 20 characters long.

The Medley System Tool now supports sysout and microcode installation using the TCP FTP protocol. This feature may be used by selecting the "TCP/FTP" device type in the main System Tool window. Update pages for the *1108 User's Guide* describing this feature, are included with the Medley release.

## 7. Input/Output

Every time you allocate space on a floppy disk that has fewer than 200 free pages, a message is printed in the prompt window. That message gives an approximate number of free pages remaining after the allocation; it's intended to give you warning when your floppy is nearing full. The page count is correct only within +/- 2 pages because the message is printed in the course of the allocation, and the floppy's directory may grow when the new file is added to it .

## 8. Machine Diagnostics

Medley Boot Diagnostics for the 1108 include changed floppy disk names and slight changes in the prompts for running diagnostics from floppies.

# 1186 User's Guide Release Notes

## What to Look For

The *1186 User's Guide* was extensively reorganized and rewritten for the Lyric Release. This section contains a summary of changes affecting 1186 environments with the Medley release. Details are described in update pages available for the *1186 User's Guide.*

In every 1186 chapter that requires use of Lisp expressions of any kind, there is a notice regarding the use of **IL:** and a suggestion that expressions, functions, and variables be typed into an Interlisp Exec.

## 1. Introduction

For Medley, the Xerox Lisp logo window has been changed to reflect the new name, Envos.

## 4. File System

Medley will accept floppy names up to 40 characters in length. Some of the Lyric font floppies have names in excess of 40 characters. Medley truncates the floppy name to 40 characters if asked to read a Lyric floppy with a longer name. The function FLOPPY.NAME is used to name a floppy. When it is not given any arguments, it returns the name stored on the floppy disk. When it is given a *NAME* argument, the floppy name is set to *NAME*. The 40 character limitation holds for both 1108 and 1186 floppies.

The {DSK} device on the 1108 and 1186 now accepts a wider range of characters in file names. Almost any character in character set 0 is acceptable. Previously, if you tried to create a file whose name included, for example, an underscore, you would see a "FILE NOT FOUND" error.

The 1108 and 1186 file systems had a problem with large partitions which would manifest itself as "HARD DISK error - can't find file page" when accessing newly created files. This would only appear on logical volumes larger than 64K pages. This problem has been fixed.

The function FILENAMEFROMID is now implemented.

## 5. Software Installation

The SKETCH executable files are too large to be contained on one floppy. The files are now distributed on two floppies: *Medley Library Floppy #3* and *Medley Library Floppy #4*. To load SKETCH, type the Interlisp exec command:

(FILESLOAD LOAD-SKETCH)

The LOAD-SKETCH function will copy all SKETCH files from *#3*; then prompt you to insert *#4*, and the remainder of the files will be copied.

## 6. System Tools

In System Tools, it is no longer necessary to execute a **Floppy Info!** command before attempting a **List!**.

---

The Medley System Tool now displays an error message when an NS Domain or Organization name is more than the allowed 20 characters long.

---

The Medley System Tool now supports sysout and microcode installation using the TCP FTP protocol. This feature may be used by selecting the "TCP/FTP" device type in the main System Tool window.  Update pages for the *1186 User's Guide* describing this feature, are included with the Medley release.

## 7. Input/Output

Every time you allocate space on a floppy disk that has fewer than 200 free pages, a message is printed in the prompt window.  That message gives an approximate number of free pages remaining after the allocation; it's intended to give you warning when your floppy is nearing full.  The page count is correct only within +/- 2 pages because the message is printed in the course of the allocation, and the floppy's directory may grow when the new file is added to it.

---

The following function applies  only to 1186 users:

( **dove.xor.cursor**   *&optional xor-p)*                                        [Function]

If no argument is given,  this function returns the current state of the 1186 cursor (nil implies an or'ing cursor,  t an xor'ing cursor).  If an argument is given, changes the state of the 1186 cursor appropriately.

## 8. Diagnostics

Medley Boot Diagnostics for the 1186 include changed floppy disk names and slight changes in the prompts for running diagnostics from floppies.

[This page intentionally left blank]