Annotate.AmigaGuide

COLLABORATORS			
	TITLE : Annotate.AmigaGuide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		November 24, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Annotate.AmigaGuide

1.1 Annotate 2.0

Welcome to Annotate 2.0 A Shareware text editor for the Amiga.

This is the third release of the Annotate and is the first version with a full ARexx port. The documention has been kept to a minumum (do to a lack of time on my part) but if you are used to using any other text editor or word processor this doc file and a little getting used to Ann should be enough for anyone. If you have any ideas for Ann please feel free to drop me a line (see About Annotate for info).

Table of Contents

1.2 Notes about Annotate 2.0

-FUNDAMENTALS

Some of the basic concepts of this text editor are far different than regular text editors.

Firstly, Annotate has a cursor separate from the selection. For instance, one could move the cursor wherever he or she pleases without losing the selection. In fact, if one was to edit the text inside the selection, the highlight would resize accordingly. As this text editor was created with programmers in mind, it has many features useful to programmers.

-Status Line

When editing text in Annotate, one would notice the current status line at the bottom.

A typical display would look like:

```
"work:Ann/AnnotateDoc"  Col: 30  Line:   85/   85  Modified Srl LS Sel
```

On the very left is the current document you are editing. Next is the current column and line display, then "Modified". If "Modified" is present, then the text has been edited since the last save, and will prompt for saving changes, if you try to quit. "Srl" means that the numeric keypad is in scroll mode. It could also read "Num" at which time you can input numbers with the keypad. "LS" which may also say "VS" tells wheather the current editor is in line selection mode or vertical selection mode."Sel" is used to indicate if there is any text currently selected. This is useful as many menu items act on a selection, and not neccesarily the cursor. Also, any error messages are conveyed on this bottom line, temporarily replacing the stats.

-Scroll Bar

But not just any scroll bar. A scroll bar should show you where you are in relation to the top and botton of text. Ann's scroll bar also shows you where your cursor, selection, and scroll stores are. Try it, you'll find it very useful.

A very important part of annotate is it's ScrollStores, these beasties are used to save your current position in the document. It is useful when programming, when trying to find the little piece of code within 100s of lines or more. All one has to do, is set a ScrollStore there, and then call that same ScrollStore when one wants to return. There are 6 different ScrollStore locations, so up to six positions may be saved. - Note: ScrollStore merely identifies vertical and horizotal scroll, not the actual text your looking at. On the scroll bar they appear as lines with a number under them to indicate which scrollstore the line represents.

-Quick keys not on the menu

Ctrl ArrowKey - Will move you to the beginning or end of the next word on the current line.

Shift ArrowKey - Will move the cursor to the top of screen, bottom of the screen, beginning of line and end of line.

One other recently added "feature" is the "shift-TAB". By holding down the shift key and hitting TAB, the cursor will delete the number of characters the current TAB is set to. This is useful for programming when one wants to move out one tab space. **Caution, this will delete characters as well as spaces.

-A word about making selections

Normally Ann has an unfilled be for a cursor but if you click down on the the cursor it will become filled in. If you drag the cursor to a new position and let go Ann will make a seltion from where you started to where you ended.

-About the close box

It should be noted that hitting the close box Ann last window (which would cause most programs to quit) ann will close the editor in the window but will not close the window if there is an editor in the external editor list. Insted it will put the first external editor into the window. However if there is not an external editor ann will quit just like most programs.

1.3 Startup

Starting from Workbench:

To start Ann from workbeach simply double click on the Ann icon in which case Ann will appear with the ASL file requester ready for you to use, or by double clicking on a project icon with Ann as the default tool in which case Ann will appear and imedetly load the file.

Starting from the Shell:

If you start ann from the shell you have the option of suppling a file list to be loaded at startup. You may also give a directory name as a file name in which case a file requester will appear with it's directory string set to the directory you specifed.

```
example:
          Ann s:starup-sequence s:user-startup
               (opens two windows and loads the files into them
          Ann s:
               (opens a window and file requester set to s:)
          Ann /
               (opens a window and file requester set the the current
               directory)
```

On startup Annotate will search for it's config files. Ann looks for three files:

1) AnnConfig
2) AnnTools
3) Default.AnnMacs

AnnConfig contains the deafault settings for window placement, rendering options, etc. It is created by use the "Save" gadget in the Config requester. Ann looks for AnnConfig in s:.

AnnTools is a list of menu items and execution information for anns tool menu that is userconfigerable by using this file. Ann looks for AnnTools in the Current directory and if it is not found there then Ann searches s:.

example: AnnTools

Default.AnnMacs is a Macro file created by using the "Save Macros" command from the menu. Ann looks for Default.AnnMacs in the Current directory and if it is not found there then Ann searches s:.

If any of these files are not present Ann will resort to it's built in default settings for that file.

When Ann has loaded all config files it will then search for the default public screen (in most cases this is the workbeach screen) and bring up any windows on that screen. Ann will also use the default fonts of the screen. Also Ann must run on a screen at least 640x200. If the default public screen is smaller this 640x200 ann will fail to open and supply an error msg that the screen is to small.

1.4 Example AnnTools

The following is my defualt "AnnTools" from s:. I also keep "AnnTools" specfic to a piticular project in the drawer with the project so that when I work on a project I have set of specalized tools for that project

This file begins with the word "ClickDos" and ends with a LF after "numberlist"

```
ClickDos
run <nil: >nil: Dos
Calculator
run <nil: >nil: calc
LMK
run <* >* LMK >* <*
KeyShow
run <* >* keyshow >* <*
Phone Log
run PhoneLog >nil: bigger:numberlist
```

1.5 Menus

Annotate uses standard Amiga menus with one exception. All quick keys are case sensitive. For example:

Menu Item	Quick Key	You Press
"Open New..."	Amiga-O	Amiga-Shift-o
"Open ..."	Amiga-o	Amiga-o

The following is a list of all Ann menu items with a breaf description for eash one.

Project

Edit

Search

View

Macros

```
Tools
```

1.6 Project Menu

```
ABOUT - This item merely give some information on this version of
        Annotate and a brief description. ( And a plea for "gifts".)

NEW - This will start a "new" document in a new window (it will be the
      front one).

OPEN NEW - This will bring up the AmigaDOS 2.04 requester for opening
           files and will open the selected one on a new screen
           placed in the front.

OPEN - This will bring up the AmigaDOS 2.04 requester for opening
       files and will open the selected one. Careful this is
       destructive, it will replace current document.

SAVE/CLOSE - This saves the file. (As it's current name - rewriting the
       old file if there was one.)  It will prompt for a name if this
       is a "new" document. Then After the file is saved the window
       and editor are closed.

SAVE ALL MODS - This saves ever file that has been modified using
          it's current name.

SAVE AS - Lets you save the file as a different name, and this will
          remain the current name.

SAVE - This just saves the file. (As it's current name - rewriting the
       old file if there was one.)  It will prompt for a name if this
       is a "new" document.

PRINT - sends the current window out prt:

LAST SAVED - This will "revert" the text to the last version saved.
             Caution - destructive!

DELETE FILE... - This will bring up an AmigaDOS 2.04 requester
                 prompting for which file to delete.
                 This is obviously DESTRUCTIVE!

SAVE ALL/QUIT - This option saves all current documents open (like
                doing a "SAVE" for each one) and quits the program.

QUIT - This option lets the user quit without saving the documents.
       If changes have been made, a requester will pop up asking if
       you wish to really quit.
```

1.7 Edit Menu

LINE/VERT SEL - The selection will change from a line selection to a vertical selection. To find out what mode the window is in, look at the status bar it will have an "LS" or "VS" in it, meaning Line Select mode or Vertical Select mode respectively.

CUT - This will allow you to "cut" (which copies to the clipboard and deletes) to one of six clipboards from the current selection.

COPY - This will allow you to "copy" (which copies to the clipboard) to one of six clipboards from the current selection...

PASTE - This will allow you to "paste" (which prints from the clipboard) from one of six clipboards to current cursor position.

ERASE - This will delete the current selection. Caution - destructive.

FOLD - This will fold the currently selected lines.

UNFOLD - This will unfold the line the cursor is currently on.

UNFOLD ALL - This will unfold all of the folded lines in the current document.

SHIFT RIGHT - This will shift all of the characters in the current selection right on the assumption that ENTIRE lines are selected.

SHIFT LEFT - This will shift all of the characters in the current selection left on the assumption that ENTIRE lines are selected.

UPPER CASE - This will shift all of the characters in the current selection to upper case.

LOWER CASE - This will shift all of the characters in the current selection to lower case.

DUPLICATE LINE - This will take the line the curent cursor is on, and will duplicate it on the next line.

MERGE LINE - Merge line takes the current line the cursor is on and tacks the next line to the end of it. (Try it and you will see what I meen.)

DELETE LINE - This menu item deletes the line the cursor is on.

LOCK LINES - Sets all lines in the current selection to locked. This makes the selected lines uneditable. You can use this featue to protect certain parts of your document. Locked lines will appear underlined if you have the "Draw Line Locks" gadget ticked in the config requester otherwise locked will appear identical to unlocked lines.

UNLOCK LINES - Sets all locked lines in the current selection to unlocked. (see LOCK LINES for more info)

START SEL - Start a selection from the keyboard. This feature is mostly here for making selections from ARexx and Macros.

END SEL - End a selection from the keyboard. This feature is mostly here for making selections from ARexx and Macros.

1.8 Search Menu

FIND... - Find produces a "find" requester. The "string" gadget accepts a string of characters to find.

FIND NEXT - Uses the string from the original find requester to find the next instance of the string.

REPLACE - Produces a "replace" requester with several option gadgets. The string gadgets accepts strings to search for and replace with. There are cycle type gadgets from search type and prompt type.

Sel->Find - Takes the current selection and places it in the "find" requester of Find or Replace.

Sel->Replace - Takes the current selection and places it in the "replace" requester.

Line Number... - Brings up a requester prompting for the line number you wish to go to.

Match Brace - This finds the opposite bracket or brace and takes the cursor to it.

Match Brace/Sel - This finds the opposite bracket or brace and selects everything between the two.

Select All - Simply selects the entire document.

Select Line - Selects entire line the cursor is on.

Select Word - Selects the word your cursor is on.

1.9 View Menu

ED EXCHANGE... - OK, this is the coolest thing in Ann. What it will let you do is have several files loaded into memory, but not necessitating have windows for them all. And, if you want to look at a file in one of the buffers, you simple exchange it for a file that is in the current window. Or you could put it in a new window of it’s own. You can also open close windows but still keep the file in memory or open files into the current window and have the current text in the window buffered off into memory. To make a long story short, (too late, I know) Exchange is the best place to open and close files and windows if you are working on a large number of files, it’s hard to explain this feature fully. So... all I can say is try learning to use it. Once you figure it out, you’ll never use the normal Open from the file menu again.

Shrink Ann - Makes Annotate shrink to minimum possible size.

Enlarge Ann - Makes Annotate grow to a maximum possible size.

Ann to Front - Takes Annotate’s window and pushes it front.

Ann to Back - Takes Annotate’s window and pushes it to the back.

Go->ScrollStore - (1 - 6) Goes to previously set scrollstore from one to six.

Set->ScrollStore - (1 - 6) Sets the "scroll store" to the current location of the cursor. (One may call the location back at any time by using the Go->ScrollStore.)

Goto Cursor - Moves your "scroll" or view back to the cursor. It may not be at the cursor if you have used the Scroll Store.

Goto Selection - Moves your "scroll" or view to the currently selected text.

Scroll/Num - Sets the keypad between scroll movement and typing numbers. Much like a "NumLock" key on an IBehM.

Regen All - This will redraw all of Annotate’s current windows. It will also set Ann Global varables named GRAPHIC_UPDATE and SHOW_REQUEST to ON (see AREXX section under SET).

1.10 Macros Menu

PLAY MACRO - Play macro is use to play a maros 1 to 10 depending on the number you select. Note that the Quick key is Shift the funtion key which coresponds to the macro you wish to play. Also if you hold down the Ctrl key in the quick key combination the macro will shut down screen update until after the macro has finished. For macros that do a lot of screen update this can speed up the macro dermaticly.

RECORD MACRO - Will start recoring a macro in the number you selected. Use ’Stop Recording’ when you are finished.

STOP RECORDING - If you Recording a Macro this will stop the recording and return Ann to it’s normal state.

OPEN MACROS - Load a macro file made with ’Save Macros’.

SAVE MACROS - Save all macros to a file.

1.11 Tools Menu

Config... - This is the item for changing the default configuration of Annotate. Almost everything here is self explanatory except for a few items. The "scroll unit" is the number of characters from all sides of the window, where Annotate will decide to scroll if you are in. (- Note: you may wish to make it quite small, if you are using a small window size. It will let you to continue to operate with scrolling intact. The "Window TopEdge" is where you want the top, left edge of Annotate to default to (in pixels from the top, left of your screen). "Window Width" and "Height" are also set the default configuration of Ann’s window in pixels. The make .info file switch, determines if icons are created for your text files. (If there is a .info file, there is an icon.) The make .bak switch sets wether or not the program creates ".bak" backup files, of your current version of the text you are editing. Auto Indent tells Annotate if it should automatically indent you when you press return to where you indented to on the previous line. You can also set the default search and prompt type for the various requesters throughout Annotate (ie. Search and Replace requesters).

RX FILENAME - After selecting this item a file requeter will apear with ’#?.rexx’ in the patten gadget. When you select a file Ann will run it as ’rx filename’. Note this will only work if you have ARexx properly installed on your system.

FILENAME TO CLIP0 - This will copy the current windows filename to unit 0 of the clipboard.

INSERT ASCII CODE - Will produce a requester letter you input a number between 0 and 255. This number is inserted at cursor position when the OK gadget is pressed.

REPEAT NEXT EVENT - Will produce a number requester. Whatever number you put into the requester will the number of times the next event is done. For example, if you input the number 10 into the requester and after hitting OK you press the letter ’j’, insted of 1 j being inserted into the editor 10 j’s will be inserted.

JUSTIFY LINE - Justify line has three options from the menu, Left, Center, and Right. Left will push the line the cursor is on to the far left. Center will center the line between the left the Wordwrap Colum (set in the config requester). Right will push the line up so it touches the Wordwrap Col.

-Tool Items - after this menu item, are the tools that may be added. Tools may be added by adding them to the text file "AnnTools. Explained previously in the starup section.

If they are added correctly, then they will appear here.

1.12 ARexx

The following are a list of base commands for the annotate ARexx port which can be addressed as:

```
address('Annotate_Rexx')
```

Each of these base commands may have more specfic commands and/or parmeters follow them. For example the if you wished to play macro 6 via the rexx port the command would be:

```
DOMENU PLAY_MACRO 6
```

In this case DOMENU is the base command, PLAY_MACRO is the first sub command, and 6 is the second sub command. Note that memu commands never take any parmeters.

An example of a command that does take parmeters is:

```
MOVE_CURSOR REL 5 8
```

Where MOVE_CURSOR is the base command, REL (for relative to the current position of the cursor) is the sub command and 5 and 8 are the parameters for deltaX and deltaY. This command will move the cursor 5 spaces to the right and 8 spaces down exatly the same way you can do with the cursor keys.

If OPTIONS RESULTS has been specficed in the rexx program then many of the following commands will also return valualbe information.

The following is a complete list of Annotates rexx commands, subcommands, parmeters and return values.

DOMENU

SET

GET

INSERT

MOVE_CURSOR

REQUESTER

Other Commands

Example Code:

Requster Demo

Seach and Load

```
      Replace And Count
```

1.13 Replace And Count

```
/*  This Rexx Macro will run through the editor replacing
the string in the find requester with a number. Each new
occruce of the find string found will be replaced with a
higher number.

ex. if 'test' was in the find string and the editor read:

          test test test test test

you could have this macro replace tests with:

          1 2 3 4 5

*/

OPTIONS RESULTS
address 'Annotate_Rexx'

count = 0

REQUESTER NUM "'Inderations?' '#' '100'"

if RC == 0 then
do
     Ids = RESULT

     REQUESTER NUM "'Startnum?' '#' '0'"
     CurNum = RESULT

     if RC == 0 then
     do
          ids = CurNum+ids
          do while ids > CurNum
               domenu find_next
               Get sel_size
               if RC ~= 0 then
                    count = ids
               else
               do
                    DOMENU Erase
                    insert string CurNum
                    CurNum = CurNum+1
               end
          end
     end
end

exit rc
```

1.14 Requster Demo

```
/* A Quick Requester Demo */

OPTIONS RESULTS
address 'Annotate_Rexx'

REQUESTER ANSWER "'Do _List' 'C_ancel' '_OK'"

say RESULT

if RESULT == '_OK' then
do
     REQUESTER List "Window_Name Ca_ncel 'This is ' a test ' of the' list gadget"
     say RESULT
end

REQUESTER NUM "'Num Request' 'Hel_lo' '7456'"

if RC == 0 then
     say RESULT
else
     say "You Hit Cancel"

REQUESTER STRING "'String Request?' 'Te_xt' '_Ok doky' '_Na'"

if RC == 0 then
     say RESULT
else
     say "You Hit Cancel"


exit rc
```

1.15 Seach and Load

```
/* This rexx program brings up a list requester with some file names.
After the user picks a file (as long as they didn't pick cancel) the
program searches to see if ann has that file in a window and if it finds
it, Ann brings the window to front. If ann dons't find it, Ann searches
the external editor list in memory and brings the editor (if it's there)
into the current window. If ann still hasn't found the file, Ann will load
it off disk and into the current window and push the editor in the current
window into the external editor list.

Whenever I'm working on a project this is the program I us to manage my
files. You can expand it simply by adding more file to the List requester.
*/

OPTIONS RESULTS

address('Annotate_Rexx')

/* find out what file the user wants */
```

```
REQUESTER LIST "'Open Ted File' '_Cancel' 's:startup-sequence' 's:user-startup'"

/* if they didn't cancel find the file they want */

if RESULT ~== '_Cancel' then
do

     userhit = RESULT

     /* is the file in a window */

     SET WRK_WIN userhit

     if RC = 0 then
     do
          domenu ann_to_front /* the file is in a window so we can stop */
          exit
     end

     /* is the file in an external editor */

     RC = 0
     count = 1
     RESULT = ""

     do while RC = 0

          GET EX_ED_NAME count

          if userhit == RESULT then
          do
               EXCHANGE_EDS count  /* the file was is an external editor
                                   so bring the editor into the current
                                   window and quit */
               domenu ann_to_front
               exit
          end

          count = count + 1

     end

     /* The file is not in memory so load it into the current window and
     put the editor which is in the current window into the exteral
     editor list */

     OPEN_EX_EDITOR userhit

end
```

1.16 domenu

The DOMENU command can be used to access most menu items by simply writting out the menu item after DOMENU. Menu items with more then one word in them such as "Last Saved" must be written as "LAST_SAVED" so ann will not think the word "SAVED" is another subcommand. DOMENU commands never take any parms and never return anything. Below is a complete list of the menu items supported be Ann. Indented items are sub commands of the previos item and must be used to make the command work.

```
          ABOUT...
          NEW
          OPEN_NEW...
          OPEN...
          SAVE/CLOSE
          SAVE_ALL_MODS
          SAVE_AS...
          SAVE
          PRINT
          LAST_SAVED
          DELETE_FILE...
          SAVE_ALL/QUIT
          QUIT
          LINE/VERT_SEL
          CUT
               0 to 5
          COPY
               0 to 5
          PASTE
               0 to 5
          ERASE
          FOLD
          UNFOLD_ALL
          SHIFT_RIGHT
          SHIFT_LEFT
          UPPER_CASE
          LOWER_CASE
          DUPLICATE_LINE
          MERGE_LINE
          DELETE_LINE
          LOCK_LINES
          UNLOCK_LINES
          START_SEL
          END_SEL
          FIND...
          FIND_NEXT
          REPLACE...
          SEL->FIND
          SEL->REPLACE
          LINE_NUMBER...
          MATCH_BRACE
          MATCH_BRACE/SEL
          SELECT_ALL
          SELECT_LINE
          SELECT_WORD
          ED_EXCHANGE...
          SHRINK_ANN
          ENLARGE_ANN
          ANN_TO_FRONT
          ANN_TO_BACK
          SET->SCROLLSTORE
               1 to 6
          GO->SCROLLSTORE
               1 to 6
          GOTO_CURSOR
          GOTO_SELECTION
          SCROLL/NUM
          REGEN_ALL
          PLAY_MACRO
               1 to 10
          OPEN_MACROS
          SAVE_MACROS
          CONFIG...
          FILENAME_TO_CLIP0
          INSERT_ASCII_CODE
          REPEAT_NEXT_EVENT
          JUSTIFY_LINE
               LEFT
               CENTER
               RIGHT

 Some examples of DOMENU
```

1.17 DOMENU Examples

```
DOMENU SET->SCROLLSTORE 5
DOMENU CUT 0
DOMENU DUPLICATE_LINE
DOMENU JUSTIFY_LINE CENTER
DOMENU CONFIG...
```

1.18 set

```
Subcommads of SET with their parms and return vaules:

WIN_STATS      Moves and resizes the current window.

     call      SET WIN_STATS WindowX WindowY Width Height

     parms     WindowX,WindowY,Width,Height are all numbers describing
               where to move and how to size the current window.

     result    "WindowX WindowY Width Height" which are all numbers
               desribing the resulting window size and position.


DEF_WIN_STATS  Sets defualt window size and position for new windows.

     call      SET DEF_WIN_STATS WindowX WindowY Width Height

     parms     WindowX,WindowY,Width,Height are numbers used as the new
               default window (see Config requester).
     result    "WindowX WindowY Width Height" which are all numbers
               desribing the resulting defualt window size and position.


WRK_WIN

     call      SET DEF_WIN_STATS WindowName

     parms     WindowName is a the name of the window you want all futher
               rexx commands to be directed towards.

     result    WindowName if the window WindowName was found.


GRAPHIC_UPDATE

     call      SET GRAPHIC_UPDATE ON/OFF

     parms     ON or OFF. If you turn GRAPHIC_UPDATE off all rendering in
               ann will be stopped making most operations speed up.
               Turning it back on will cause a total refresh.

     result    The status of GRAPHIC_UPDATE, namely ON or OFF.


SHOW_REQUEST

     call      SET SHOW_REQUEST ON/OFF

     parms     ON or OFF. If you turn SHOW_REQUEST off all requester in
               Ann will automaticlly return as if the OK gadget was
               selected without even showing the requester to the user.

     result    The status of SHOW_REQUEST, namely ON or OFF.


FIND_TYPE      Used to set the Search Type gadget in the search requesters.

     call      SET FIND_TYPE TypeNum

     parms     TypeNum is a number from 0 to 2 which mean:
               0 Case Sensative
               1 Non-Case Sensative
               2 Wild Cards

     result    The old FIND_TYPE TypeNum.


REPLACE_PROMPT_TYPE Used to set the Prompt Type gadget in the search requesters.

     call      SET REPLACE_PROMPT_TYPE TypeNum

     parms     TypeNum is a number from 0 to 2 which mean:
               0 Always Prompt
               1 No Prompt
               2 Prompt Once
     result    The old REPLACE_PROMPT_TYPE TypeNum.


FIND_TEXT      Used to set the Find string gadget in the search requesters.

     call      SET FIND_TEXT String

     parms     String is a string that you want in the string gaget.

     result    The old FIND_TEXT String.


REPLACE_TEXT   Used to set the Replace string gadget in the search requesters.

     call      SET REPLACE_TEXT String

     parms     String is a string that you want in the string gaget.

     result    The old REPLACE_TEXT String.


STAUS_LINE     Used to set the current windows status line.

     call      SET STAUS_LINE String

     parms     String is a string that you want written into the current windows
               status line.

     result    none.


DRAWER         Used to set the Drawer string gadget in the file requester.

     call      SET DRAWER String

     parms     String is a string that you want in the string gaget.

     result    The old DRAWER String.


PATTERN        Used to set the Patten string gadget in the file requester.

     call      SET PATTERN String

     parms     String is a string that you want in the string gaget.

     result    The old PATTERN String.


SEEK_LINE      Used to set the LineNum gadget in the seek LineNum requester.

     call      SET SEEK_LINE Num

     parms     Num is a num that you want in the num gaget.

     result    The old SEEK_LINE num.
```

```
ASCII_CODE     Used to set the Ascii code gadget in the insert ascii code  ↩
   requester.

     call      SET ASCII_CODE Num

     parms     Num is a num that you want in the num gaget.

     result    The old ASCII_CODE num.
```

1.19 get

```
Subcommads of GET with their parms and return vaules:

SCR_SIZE       Used to get the size of the screen.

     call      GET SCR_SIZE

     result    "Width Height" of the screen Ann is running on.


WIN_STATS      Used to get the current window size and position.

     call      GET WIN_STATS

     result    "WindowX WindowY Width Height" which are all numbers
               desribing the current windows size and position.


DEF_WIN_STATS  Used to get the default window size and position.

     call      GET DEF_WIN_STATS

     result    "WindowX WindowY Width Height" which are all numbers
               desribing the default window size and position.


SEL_TEXT       Used to get the current selection.

     call      GET SEL_TEXT

     result    A string contaning the current selection or an err if there
               is no selection.


CHR            Used to get the character under the cursor

     call      GET CHR

     result    "char ascii CursorX CursorY" where char is the character
               under the cursor, ascii is the ascii value of char and
               CursorX and CursorY is the position of the cursor.


WIN_NAME       Used to get the name of a window
```

```
     call      GET WIN_NAME WindowNum

     parms     WindowNum is a number of the window you want the name of or
               it can be 0 if you want the name of the current window.

     result    A string contaning the window name or an err if the
               WindowNum is higher then the number of windows open.


SEL_SIZE       Used to get the start and end points of the selection

     call      GET SEL_SIZE

     result    "MarkStartX MarkStartY MarkEndX MarkEndY" which are all
               numbers describing the cursor pointions of the selection
               start and end. If there is no selection an err will be
               returned.


EX_ED_NAME     Used to get the name of a external editor

     call      GET EX_ED_NAME EdNum

     parms     EdNum is a number of the editor you want the name of.

     result    A string contaning the editor name or an err if the
               EdNum is higher then the number of External editors open.
```

1.20 insert

```
Subcommads of INSERT with their parms and return vaules:

STRING         Used to insert a string of text at the cursor position.

     call      INSERT STRING String

     parms     String is the string you want inserted.

     result    "TRUE" if all went well.


RETURN         Used to insert LFs at the cursor position.

     call      INSERT RETURN Num

     parms     Num is the number of returns to insert.

     result    "TRUE" if all went well.


SPACE          Used to insert spaces at the cursor position.

     call      INSERT SPACE Num
     parms     Num is the number of spaces to insert.

     result    "TRUE" if all went well.
```

1.21 move_cursor

Subcommads of MOVE_CURSOR with their parms and return vaules:

```
REL            Used to move the cursor some distence form where it is.

     call      MOVE_CURSOR REL DeltaX DeltaY

     parms     DeltaX and DeltaY are numbers which are used as offsets to
               move the cursor from it’s current position.

     result    "DeltaX DeltaY X Y" where DeltaX and DeltaY are the actual
               distence the cursor was able to move and X and Y are the
               cursors new absolute position.


ABS            Used to move the cursor to an absolute position.

     call      MOVE_CURSOR ABS X Y

     parms     X and Y are numbers which are used as absolute positions to
               move the cursor to.

     result    "DeltaX DeltaY X Y" where DeltaX and DeltaY is the distence
               the cursor moved and X and Y are the actual new cursors
               absolute position.
```

1.22 requester

Subcommads of REQUESTER with their parms and return vaules:

```
ANSWER         Offers the user one of three choies.

     call      REQUESTER ANSWER Text Cancel OK Mabye

     parms     Text is a string and the question being asked.
               Mabye, Cancel and OK are strings for up to three gadgets (
               you don’t have to use them all).

     result    A string which contains the option chosen.


LIST           Offers the user one of many choices and a cancel.

     call      REQUESTER LIST WindowTitle CancelText Option1 Optoin2 ...

     parms     WindowTitle is the window title and should be the question
```

```
               begin asked.
               CancelText is a string for the Cancel gadgets text.
               Option1 ... are the option of the list.

     result    A string which contains the option chosen. This option
               could be any of the Option texts or the Cancel text.

NUM            Lets the user input a number.

     call      REQUESTER NUM WindowTitle NumGadText DefaultNum

     parms     WindowTitle is the text put into the window title.
               NumGadText is the text put beside the num gadget.
               DefaultNum is the number the requester comes up with.


     result    The number the user selected or an err if they canceled.


STRING         Lets the user input a number.

     call      REQUESTER STRING WindowTitle StringGadText OKGad CancelGad

     parms     WindowTitle is the text put into the window title.
               StringGadText is the text put beside the string gadget.
               OKGad is the text used for the OK gadget.
               CancelGad is the text used for the Cancel gadget.

     result    The string the user input or an err if they canceled.
```

1.23 Other Commands

```
ANN_VERSION    To find out what version of Ann you're talking to.

     call      ANN_VERSION

     result    A string containg Annotates version.

DEL            Deletes characters from the cursor postion.

     call      DEL Num

     parms     Num is the number of characters to delete.

     result    "TRUE" if all went well.


OPEN_IN_WIN    Loads a file into the current window

     call      OPEN_IN_WIN FileName

     parms     FileName is a string with the name of the file to be loaded

     result    none.
```

```
OPEN_EX_EDITOR Loads a file into the current window and puts the editor in
               the current window into the external editor list.

     call      OPEN_EX_EDITOR FileName

     parms     FileName is a string with the name of the file to be loaded

     result    none.


OPEN_WIN       Opens a window and puts the first editor in the external
               editor list into it.

     call      OPEN_WIN

     result    none.


CLOSE_BOTH     Closes the current window and the editor in it. Same a
               clicking the close gadget on the window.

     call      CLOSE_BOTH

     result    none.


CLOSE_WIN      Closes the current window and puts its editor into the
               external editor list.

     call      CLOSE_WIN

     result    none.


CLOSE_ED       Closes the editor in the current window and puts its puts
               the first editor from the external editor list into it.

     call      CLOSE_ED

     result    none.


EXCHANGE_EDS   Exchages an external editor with the editor in the current
               window.

     call      EXCHANGE_EDS Num

     parms     Num is the number of the extrenal editor.

     result    none if the Num was valid else an err is returned.
```

1.24 CopyRight Stuff

ANNOTATE

by

DOUG BAKEWELL

version 2.0 Aug 20,1993

This text editor is SHAREWARE and may be freely copied
as long as all files are included and unaltered.

If Annotate is of use to you please send a gift (say $30 or so) to:

Doug Bakewell
528 Tait ST.
Victoria B.C.
Canada
V8Z 2C8

DISCLAIMER OF WARRANTY AND LIABILITY

This software and manual file are offered "as is" and without warranties as to performance or merchantability. Any statments made about the uses of this software do not constitute warranties and shall not be relied on by the user in deciding to use this program.

This program is given without any express or implied warranties whatsoever. Because of the diversity of conditions and hardware under which this program may be used, no warranty or fitness for a particular purpose is offered. The user is advised to test the program thoroughly before relying on it. The user must assume the entire risk of using the program.