J 6 -- Contents

Session Manager

The session manager window contains an input/output log, an input window, and a status line.

TAB switches between the i/o log and the input window.

Up and down arrow keys recall input.

SHIFT+DEL cuts to the clipboard. CTRL+INS copies. SHIFT+INS pastes.

CTRL+BREAK interrupts execution.

CTRL+D and ENTER end session.

Language Summary

		.	:
=	Self-Classify . Equal	Is (Local)	Is (Global)
<	Box . Less Than	Floor . Lesser of	Decrement . Less Or Equal
>	Open . Larger Than	Ceiling . Larger of	Increment . Larger Or Equal
_	Negative Sign / Infinity	Indeterminate	Infinity
+	Conjugate . Plus	Real/Imag . GCD (Or)	Double . Not-Or
*****	Signum . Times	Polar . LCM (And)	Square . Not-And
–	Negate . Minus	Not (1-) . Less	Halve . Match
%	Reciprocal . Divide	Matrix Inv . Mat Divide	Square Root . Root
^	Exponential . Power	Natural Log . Log	Power . Chain
$	Shape Of . Shape	Suite	Self-Reference
~	Both . Cross	Nub .	Nub Sieve . Not-Equal
|	Magnitude . Residue	Reverse . Rotate	Transpose
.	Det . Dot Product	Even	Odd
:	Explicit Definition	Obverse	Adverse
,	Ravel . Append	Ravel Items . Apnd Items	Itemize . Laminate
;	Raze . Link	Cut	Word Formation .
#	Tally . Copy	Base 2 . Base	Antibase 2 . Antibase
!	Factorial . Out Of	Fit (Customize)	Foreign
/	Insert . Table	Oblique . Key	Grade Up . Sort
****	Prefix . Infix	Suffix . Outfix	Grade Down . Sort
[	Same . Left	Lev	Same
]	Same . Right	Dex	Same
{	Catalog . From	Head . Take	Tail .
}	Amend	Behead . Drop	Curtail .
"	Rank	Do . Do Left If Error	Format
`	Gerund		Evoke Gerund
@	Atop	Agenda	At
&	With / Compose	Under (Dual)	Appose
?	Roll . Deal		
)	Label	**a.** Alphabet	**A.** Atomic Permute
b.	Boolean	**c.** Characteristic	**C.** Cycle-Direct . Permute
e.	Raze In . Member (In)	**E.** . Member of Interval	**f.** Fix
i.	Integers . Index Of	**j.** Imaginery . Complex	**NB.** Comment
o.	Pi Times . Circle	**p.** Polynomial	**r.** Angle . Complex
x.	Left Argument	**y.** Right Argument	**0:** Zero
			1: One

!: Enumeration

```
0!:0   Host
0!:1   Host (No Result)
0!:2   Script
0!:3   Silent Script
0!:4   Lines
0!:5   Silent Lines
0!:55  Terminate Session

1!:1   File Read
1!:2   File Write
1!:3   File Append
1!:4   File Size
1!:11  File Indexed Read
1!:12  File Indexed Write
1!:55  File Erase

2!:0   WS Name Class
2!:1   WS Name List
2!:2   WS Save
2!:3   WS PSave
2!:4   WS Copy
2!:5   WS PCopy
2!:55  WS Erase Object

3!:0   Internal Type
3!:1   To    Internal Representation
3!:2   From Internal Representation

4!:0   Name Class
4!:1   Name List
4!:55  Erase
4!:56  Erase Given Name

5!:0   Fix
5!:1   Atomic   Representation
5!:2   Display Representation
5!:3   String   Representation
5!:4   Tree     Representation
5!:5   Linear   Representation

6!:0   Time Now
6!:1   Time Since Start
6!:2   Time to Execute
6!:3   Delay
7!:0   Space Now
7!:1   Space Since Start
7!:2   Space to Execute

9!:0   Query Random Link
9!:1   Set    Random Link
9!:2   Query Display Representations
9!:3   Set    Display Representations
9!:4   Query Prompt
9!:5   Set    Prompt
```

9!:6   Query Box Drawing Characters
9!:7   Set   Box Drawing Characters
9!:8   Query Error Messages
9!:9   Set   Error Messages

10!:k (LinkJ) C Function

11!:0 Window Driver
11!:1 Windows Visual Edit

128!:0 QR Decomposition
128!:1 R Inverse

Window Driver

`11!:0` is the Window Driver interface.   For example:

`11!:0 'pc abc; pshow;'`
Create a window named abc.
`11!:0 t=.'pc asdf; xywh 10 10 50 15; cc "press me" button; pshow;'`
Create a window named asdf with a button labelled "press me".
`11!:0 'reset;'`
Close windows.

See <u>Window Driver Reference</u> for further details.

`11!:1` is a visual editor.   For example, the phrase `11!:1 t` edits the asdf window defined above.
See <u>Visual Edit</u> for further details.

Files in the wp\ subdirectory illustrate various aspects of the interface.   To execute these files, proceed as follows:

```
eof  =. 26{a.
read =. 1!:1
wd   =. 11!:0
wp   =. wd@(-.&eof)@read

wp <'\j\wp\checkbox.wp'
wp <'\j\wp\listbox.wp'
...
```

Window Driver Reference

The Window Driver lets you program the graphical user interface.

A window program is a character vector of statements that create, manipulate, and interact with windows.

'pc abc; pshow;' has two statements. The first statement is the command pc with parameter abc and the second statement is command pshow.

Statements must end with semicolon.

Commands and parameters are separated by whitespace. Whitespace is 1 or more characters from the set (space, carriage return, line feed, tab, vertical tab, and form feed).

Simple parameters start with an alphanumeric or - and run to the ; or whitespace.

Delimited parameters start with " or AV255 and run to the matching delimiter. AV255 is the last character of the set of 256 characters.

'pc "my window";pshow sw_showmaximized ;' has two statements: pc command with parameter "my window" and pshow command with parameter sw_showmaximized.

Function wd executes a window program.

Parent

The pc command creates a parent window that is initially not visible. It can be displayed in various ways with the pshow command. A parent window is the display area for the menus and child windows required by a user interface.

There can be more than one parent window. Commands affect the selected parent.

The pc command selects the new parent and the psel command can select any parent.

The pco command creates a parent window that is 'owned' by the currently selected parent. The owner is disabled until the new parent is closed.

Child

The cc command creates a child window that belongs to the selected parent.

A child window is often called a control.

There can be more than one child for a parent. Commands affect the selected child of the selected parent.

The cc command selects the new child and the csel command can select any child of the selected parent.

The location and size of a child in a parent is set with the rectangle.

The xywh command sets the rectangle;

The adj and other commands adjust the values of the rectangle.

The rectangle is defined by 4 integers:

- x units from the left
- y units from the top
- w units in width (to the right)
- h units in height (to the bottom)

Units are in terms of the average character size in the system font.

x and w values are 1/4 of the width and y and h values are 1/8 of the height of the system font. For example, a control with xywh values of 40 80 100 160 would be: 10 characters from the left edge of the parent; 10 lines down from the top edge of the parent; 25 characters wide; and 20 lines high.

Location and size defined relative to the size of the system font allow window definitions to display reasonably on different screens at different resolutions.

A control with class isigraph is a graphbox in which graphs can be drawn.

The coordinate system is: x values from 0 at the left to 1000 at the right, and y values from 0 at the bottom to 1000 at the right.

The wait command waits for user input. A pushbutton, a menu item, a function key, an accelerator key, or closing a window end the wait. The wait must be the last command in the window program.

The wd result is an n by 2 array of boxed character vectors. The first column is the id of the data.

System ids start with *. *type is the type of result, *error is the error that occurred, and *index is the 0 origin index in wp of the command in error.

User ids are set in pc and cc commands.

Ids are lowercase. NOTE: cc Aa button; labels the button as Aa, but the id is aa.

The wd result first row is *type and indicates.how it ended. *type values are:

1 error	2 nowait	3 fkey	4 menu	5 button
6 close	7 enter	8 cancel	9 break	

Visual Edit

Function wdvedit allows visual editing of a wp that defines a parent window and its controls. The argument is the wp to edit and the result is a modified wp with new values for the xywh commands.

wdvedit displays the parent and its controls in a disabled state and allows the mouse to move and resize marked rectangular areas.

The marked areas are the rectangles defined by the xywh commands in the wp. The tab key cycles through the xywh rectangles. The mouse can grab an area and move it and can grab an edge or corner to resize it. When the mouse button is released the wp is rerun and the new result is displayed.

Moving or resizing an area results in new values for the xywh command in the wp. Moving and resizing a marked area results in moving and resizing all controls based on that xywh command.

The marked area can be moved beyond the right or bottom edges of the parent. If the wp ends with a pas command, the parent will resize to hold the new locations of the controls.

A separate window displays xywh values of the marked area. If no area is marked then it displays the xywh values of the parent.

Fonts

cf... commands select a font.

      cfsys   selects the system proportional pitch font. This is the default.

      cfoem   select the OEM fixed pitch font.

      cfapl   selects the APL True Type font. APL input is NOT supported in a combobox.

CR and LF

Text delimited by CR is treated as if it were delimited by CR,LF.

Data read from an edit window with multiple lines will contain CR,LF pairs.

Accelerator Keys

& in the name of a button or menu item sets a keyboard accelerator. The & is not displayed and the next character is underlined. ALT + the character is the same as clicking on the button or menu item.

TAB and SHIFT+TAB cycle the focus through the children in the order they were created.

Cursor keys cycle through the controls in a group. By default controls are created as part of a group consisting only of themselves.

The group style creates a control that is grouped with the previous control. For example: 'cci b1 group;'.

The first letter of a command usually indicates a general category:

p   parent window
c   child window
g   graphics
q   queries that return information

Parameters are integer, id, text, or style.

-1 usually indicates the value is inherited from a previous value.

An id identifies a parent, child, or menu item. It is lowercase and is truncated to 31 characters.

A style parameter is a keyword. A prefix usually indicates what it applies to. For example, bs_autoradiobutton is a button style.

COMMANDS

adj i j ;   adjust rectangle X and Y values by i and ,j

adjh i ;   adjust rectangle Y value by H and i

adjw i ;   adjust rectangle X value by W and i

beep i ;   beep i times

cc id class [styles] ;

    create child. id identifies the child in results and is used to select the child as a target for other
    commands. class must be a valid class (e.g., button). 0 or more styles customize the control. id is
    used as initial name (see cn command).

ccheck b ;   check or uncheck button

cci id [styles] ;   create child with class and initial style inherited from last cc command.

cdir filespec [styles] ;

    add file list to listbox or combobox. Styles are from the set (archives, directories, drives, exclusive,
    hidden, normal, readonly, system). exclusive excludes normal files and lists only those indicated.

cenable bool ;   enable or disable child

cfapl ;   apl font

cfocus ;   set input focus

cfoem ;   oem fixed pitch font (standard pc character set)

cfsys ;   system variable pitch font

clscroll n;   scroll edit text n lines. n can be signed.

cn text ;   child name set to text

csel id ;   select child id as target for commands

csendm i j k ;   sendmessage i, wParam j, and lParam k

csendmstr i j text ;   sendmessage i, wParam j, and lParam addressing text

csetsel i ;   set selection to line i in listbox or combobo

ctext text ;   write new text in listbox, combobox, or edit box

gbrush ;   graphics select solid brush in color from last grgb command

gclear ;   clear graphics box

gline i j ;   draw line from current position to point i,j and update current position

glines i pts ;   draw connected lines. i is the number of points and pts is 2 or points

gpen i [style] ;

   select pen. pen is color from last grgb command and is i units wide. style is from the set (ps_solid, ps_dash, ps_dot, ps_dashdot, ps_dashdotdot, ps_null, ps_insideframe)

grect i j k l ;   draw rectangle at i,j,k,l with the current pen and brush

grgb red green blue ;

   select color. values are from 0 to 255. 0 0 0 is black, 255 255 255 is white, 255 0 0 is red.

gshow ;   display graph data. graph data is not displayed as it is created

gtext i j text ;   draws text in graph box at point i,j

mb title text [styles] ;

   messagebox with styles from set (mb_arbortretryignore, mb_defbutton2, mb_defbutton3, mb_iconasterisk, mb_iconexclamation, mb_iconhand, mb_iconinformation, mb_iconquestion, mb_iconstop, mb_ok, mb_okcancel, mb_retrycancel, mb_yesno, mb_yesnocancel)

menu id text ;   add menu item

menupop text;   add popup menu item

menupopz ;   ends popup menu and drops down a level

menusep ;   separator line in a popup menu

pas i j ;   parents size adjusted to provide i and j margins beyond children

pc id ;   parent create. Initially hidden. id is caption (see pn command).

pcenter ;   center parent on screen

pclose ;   close parent

pcloseok ;   allow user to close window even if wd is not waiting

pco id ;

    parent create with selected parent as owner. Previous parent is disabled to prevent user interaction until dialog with the new parent is finished. Closing enables the owner.

penable bool ;   enable/disable parent

phide;   hide parent, pshow can reshow

pmove x y w h ;

    move and resize parent. Values are in logical units and are relative to the top left corner of the screen. Value of 1 inherits current value.

pn text ;   name for parent window caption

pscale ;   parent window sizing scales size of children

psel id ;   select parent id to be target for subsequent commands

pshow [style] ;

    style is from the set (sw_hide, sw_minimize, sw_restore, sw_show, sw_showmaximized, sw_showminimized, sw_showminnoactive, sw_showna, sw_shownoactivate, sw_shownormal). sw_shownormal is the default.

qc ;   returns *children as the child ids of current parent.

qd ;   returns children ids and values for the selected parent

qp ;   returns *parents as the parent ids

qs ;   returns *styles as the styles for the cc command

rem text ;   remark

reset ;   close all windows

selno;   close does not reselect target parent and child   from when window was created

selrest;   select parent and child set by previous selsave

selsave;   save selected parent and child for subsequent restore by selrest

vedit ;   visual editing of the window defined by the commands that follow

wait ;   wait for user input

winexec text ;   execute DOS or WINDOWS program

xywh x y w h ;   sets X,Y,W,H rectangle. -1 value inherits the previous value.

xywhn ;   same as xywh, but not processed by vedit command.

Class

The cc command creates a child window with a class and <u>style</u>.

The class sets the basic appearance and behavior of the window.

Class is from the set (button, combobox, edit, listbox, static, <u>isigraph</u>)

A combobox is a combination of a listbox and an edit or a static.

The cc command creates a child window with a <u>class</u> and style.

Styles customize the appearance and behavior of the class.

Styles start with a class prefix. For example, bs_ for button styles. Styles should only be used with their corresponding class.

Styles that start with ws_ can be used with all classes.

group style groups the control with the previous control. Tab and Cursor keys recognize groups.

bs_autocheckbox
bs_autoradiobutton
bs_defpushbutton
bs_groupbox   labeled rectangle used to group buttons
bs_lefttext   checkbox and radiobutton text displayed on the left

cbs_autohscroll   horizontal scrolling
cbs_dropdown   listbox   displayed when dropdown icon is selected
cbs_dropdownlist   edit box is static and can only be a listbox selection
cbs_sort   sorted

es_autohscroll   horizontal scrolling
es_autovscroll   vertical scrolling
es_center   centers text in a multiline edit
es_lowercase   input converted to lowercase
es_multiline   multiline edit control
es_password   input displays as
es_right   aligns multiline text flush right
es_uppercase   input converted to uppercase

lbs_sort   listbox items are sorted

ss_blackframe
ss_blackrect
ss_center
ss_grayframe
ss_grayrect
ss_leftnowordwrap
ss_noprefix
ss_right
ss_simple
ss_whiteframe
ss_whiterect

ws_border   border
ws_disabled
ws_hscroll   horizontal scroll bar
ws_thickframe   thickframe border allows resizing
ws_vscroll   vertical scroll bar