

AD-A089 322

WHARTON SCHOOL PHILADELPHIA PA DEPT OF DECISION SCIENCES F/6 9/2

WINDOW MANAGEMENT SYSTEM USER'S MANUAL.(U)

AUG 80 D OPPENHEIM, A GREENFIELD, E & HURST

N00014-75-C-0440

UNCLASSIFIED

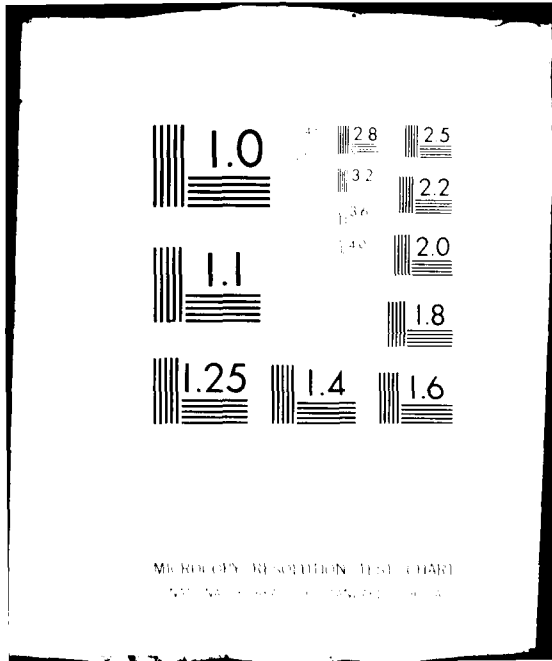
80-08-06

NL

1 of 1
AD-A089 322



END
DATE
FILMED
10-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NBS 1963-A

8

AD 10110 102



**decision
sciences**

The **Wharton** School
University of Pennsylvania



Window Management System
User's Manual

David Oppenheim
Arnold Greenfield
E. Gerald Hurst, Jr.

80-08-06

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104

Research supported in part by the Office of Naval Research
under Contract N00014-75-0440.

This document is intended
for public release and its
distribution is unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 80-08-06	2. GOVT ACCESSION NO. AD-A087 322	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) WINDOW MANAGEMENT SYSTEM USER'S MANUAL		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David Oppenheim Arnold/Greenfield E. Gerald/Hurst, Jr		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0440
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Decision Sciences The Wharton School/Univ. of Pennsylvania Philadelphia, PA 19104		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task NR049-360
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Information Systems Arlington, VA 22217		12. REPORT DATE August 1980
		13. NUMBER OF PAGES 18
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) WINDOWS, channels, information storage areas, viewports, COMINFO		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Window Management System (WINDOWS) is designed to overcome certain limitations on computer decision aids which stem from the fact that the user of such aids is typically restricted to interacting with one computer program at a time. The decision maker has access to a variety of different sources of information, such as books, graphs, maps, and notes. WINDOWS makes it possible to divide a single computer terminal into multiple "sub-terminals," thus enabling decision makers to interact with more than one computer process at a time from a single terminal. In addition, WINDOWS allows the decision maker to control all of the computer processes from a single keyboard.		

DD FORM 1473
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

processes from a single keyboard.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1.0 INTRODUCTION

The Window Management System (WINDOWS) is designed to overcome certain limitations on computer decision aids which stem from the fact that the user of such aids is typically restricted to interacting with one computer program at a time. The importance of this restriction is obvious when the use of a computer decision aid is contrasted to a more conventional decision making environment. Here, the decision maker has access to a variety of different sources of information, such as books, maps, graphs, and notes jotted down earlier in the process. Moreover, in the conventional decision making environment, "inputs" and "outputs" can be physically brought together in many combinations: a map showing the geographical disposition of ships may be viewed alongside a table of cruising ranges for different kinds of ships, despite the fact that these two inputs are from different sources.

By contrast, a person interacting with a computer decision aid generally has only serial access to information provided by the aid, especially when the decision aid is being run from a CRT type terminal, where the request for new information usually results in the destruction of data currently being displayed. Also, unless the decision maker has access to more than one computer terminal, he or she is confined to the resources of the decision aid currently running. WINDOWS helps users of computer decision aids approximate the flexibility of the conventional decision making environment by making it possible to effectively divide a single computer terminal into multiple "sub-terminals", each one having virtually all of the capabilities of a standard terminal. This enables decision makers to interact with more than one computer process at a time from a single terminal, which is useful both in contexts where the decision maker combines the resources of multiple decision aids in executing a single task, and in contexts where the decision maker executes multiple tasks simultaneously. In addition to bringing together the various information displays on a single terminal screen, WINDOWS allows the decision maker to control all of the computer processes from a single keyboard. Even where several terminals were available to an individual decision maker, WINDOWS might have the advantage of providing a better human-machine interface by keeping the person's attention focussed on a narrower, more manageable field.

ACQUISITION FOR	✓
THIS GARD	
LOS TAB	
Unannounced	
Justification	
By	
REPRODUCTION	
Approved by	
A. J. ...	
List	
A	

1.1 System Features

WINDOWS is designed to provide a software environment which enhances the user interface to a wide variety of computer decision aids. To this end, the system has been constructed to maximize flexibility, while at the same time addressing the needs of different kinds of users. Some notable design features of WINDOWS are described briefly here:

1.1.1 Functional Transparency -

There is no difference from the user point of view between running a program through a WINDOW sub-terminal and running the same program from a standard terminal. All commands directed at the program in question have the same effect they would have if the program were running in a conventional environment. For example, a CONTROL-C cancels only the program at which it is directed, and has no effect on programs running in other sub-terminals.

1.1.2 Macro Commands -

The PROFILE feature (section 3.5.1) enables users to issue a whole series of WINDOW commands by referencing a single previously created command file. This option benefits experienced users by providing for the abbreviation of frequently repeated command sequences (such as sub-terminal definitions); but it also benefits occasional users, or users who are computer-shy, by making it possible for them to use command files created by more experienced users, and thereby take advantage of the WINDOW approach without having to master its details.

1.1.3 Dynamic Sub-terminal Configuration -

Because the number, size, and other operating characteristics of sub-terminals may be changed at any time, experienced users can maintain an optimal human-machine interface as the nature of computer decision support available changes during the course of a task.

1.1.4 Program Control Of WINDOWS -

Since WINDOW commands can be issued by a program as well as by the user, designers of decision support systems can build the WINDOW approach into their aids without significant increases in programming costs. Similarly, existing decision aids can often be modified at low cost to take advantage of the multiple display capabilities of WINDOWS. In either case, the designer is free to divide responsibility for control of the user interface between decision aid and user in any way which maximizes the effectiveness of the aid.

1.2 Implementation

WINDOWS has been implemented on a DECsystem-10 running under the TOPS-10 operating system. Although the present implementation supports only Concept 100 terminals, forthcoming versions will support a wider range of terminals.

1.3 Basic Concepts

In order to be able to take full advantage of the features described below, the user should understand the three basic components of the Window Management System. These are channels, information storage areas, and viewports. Each of these is described briefly below; their interactions are explained in somewhat more detail in sections 3.1 and 4.

A channel is a line of communication to the computer operating system or monitor. All communication with the operating system takes place through a channel, including requests to log in and initiate programs, as well as the transmission of data back and forth between user and computer program. Since each channel can handle only one program at a time, and since each terminal ordinarily communicates through one channel at a time, a regular terminal user is normally restricted to interacting with one computer program at a time. To overcome this limitation, WINDOWS enables the user to create multiple simultaneously active channels and communicate through all of them from a single terminal.

An information storage area (info area) may be thought of as a message board upon which user and computer program each write messages to be read by the other. Info areas have a number of user-specified characteristics, such as the length of each line, the character set, and whether or not data can be entered from the terminal keyboard. In order to exchange data with a program, an info area must be connected to a channel, and once this connection is established, the program (as well as the computer operating system) treats the info area precisely as it would a conventional terminal. The most recently written 30 lines are retained in the info area.

A viewport is a section of the terminal screen which functions as a "window" through which the user may view the contents of an info area. User-specified properties of viewports include dimensions, location on the terminal screen, and border.

Under ordinary operating conditions, channels, info areas, and viewports are configured together into what may be thought of as "sub-terminals", each of which consists of a channel which is connected to an info area which in turn is connected to a viewport. Although this configuration is the most common, the overall flexibility of WINDOWS is increased by having the user define each of the components separately. The absence of arbitrary restrictions on the configuration of WINDOW components makes it necessary to keep in mind certain additional considerations regarding the interaction among the basic components. These considerations are presented in section 3.1 below.

2.0 RUNNING WINDOWS

In order to run decision aids or other computer programs through WINDOWS, you must log in and enter whatever commands are necessary to execute WINDOWS. Information on how to log in and execute programs is available from the installation at which you are running.

At start-up time, you are automatically provided with a sub-terminal through which to communicate with the Window Management System. This "command" sub-terminal consists of an info area named COMINFO which is connected to the channel through which WINDOWS was executed, and to a viewport named COMVIEW. COMINFO is initially defined as having 80-character ASCII lines, and as accepting keyboard input. COMVIEW is initialized as 80 characters wide and 5 lines high, with its upper left-hand corner located at the extreme upper left-hand corner of the terminal screen (i.e. 1,1). Thus, when you initiate WINDOWS, the screen will go blank and, until you command otherwise, subsequent keyboard input will appear only in the top five lines of the screen, along with WINDOWS responses to those commands. Although you are free to alter the characteristics of this command sub-terminal, you should keep in mind that all communication with WINDOWS will take place through it.

3.0 WINDOW COMMAND LANGUAGE

All commands to the Window Management System must be entered into the command info area, called COMINFO (see 2.0). To return to COMINFO from another info area, enter CONTROL-G as the first character of a new line. COMMAND keywords are printed below in capital letters, and may be abbreviated as long as the abbreviation is unique. Lower case items which are enclosed in <corner brackets> are place-holders for user-supplied values. When a group of items is enclosed in {braces}, then one of the items must be chosen either explicitly by the user, or as a DEFAULT. All items in a group which is enclosed in (parentheses) are optional. User-supplied names of channels, info areas, and viewports must be unique, and may be any combination of up to 20 ASCII characters.

WINDOWS responds in one way or another to every command entered into COMINFO. If for any reason the command cannot be executed, an error message is displayed indicating the nature of the problem. If the command is executed successfully, this will also be made clear either through a change in what appears on the screen, or by the display of

message which explicitly acknowledges that the command has been executed.

3.1 Sub-terminal Configuration

Typically, your first communication with WINDOWS will be directed at establishing the "user" sub-terminals through which to run the various programs you wish to run. For each program you wish to run through WINDOWS, you must create each of the three sub-terminal components, and then link them together. If you frequently run the same configuration of sub-terminals, you will probably want to use the PROFILE feature to establish the configuration at the beginning of each window session. See section 3.5.1 for a description of the PROFILE command. The commands required to establish a sub-terminal are described below, more or less in the order in which you would use them. The order is subject only to common sense restrictions, such as that a channel and an info area must both be defined before they can be connected. There are in addition a few simple guidelines which should be observed to avoid undesirable results.

Guidelines for sub-terminal configuration

1. The width of an info area should be at least as great as the width of the viewport to which it is CONNECTed.
2. The dimensions and locations of viewports should be specified so that no two viewports "overlap", i.e., so that no two viewports share the same area of the terminal screen, even partially.
3. No viewport should be defined so that any part of it lies off the terminal screen.

With the exception of the last, these guidelines do not constitute hard and fast restrictions. They may be safely ignored provided that the user understands certain additional features of the interactions among WINDOW components. These further considerations are explained in section 4. On the other hand, these further considerations can be ignored by users who are willing to abide by the above guidelines.

Associated with each of the sub-terminal components is a DEFINE command which causes WINDOWS to create an entity of the appropriate type with the characteristics specified in

the command string. The three DEFINE commands are described here.

3.1.1 Defining A Channel -

```
DEFINE <channel name>
```

This command creates a channel with the name indicated. Channels have no user-defined characteristics. For example, the command

```
DEFINE channel-1
```

establishes a channel named 'channel-1'.

3.1.2 Defining An Info Area -

```
DEFINE INFORMATION <info area name>  
  {READWRITE, READONLY}  
  WIDTH <w> CHARACTERS  
  {ASCII, APL}  
  END
```

This command creates an info area with the name indicated, and with the operating characteristics specified, where

If READONLY is specified, the info area will not accept terminal input.

w may have any value from 1 to 132.

If APL is specified, the terminal character set will be changed to APL whenever input is being directed to this info area.

END signals the end of an info area definition.

For example, the set of commands:

```
DEFINE INFORMATION info-area-1  
  WIDTH 40 CHARACTERS  
  END
```

sets up an info area named info-area-1 that retains the most recently entered 30 lines, each of which is 40 characters wide. Since READONLY has not been specified, the default READWRITE is in effect, so info-area-1 will accept keyboard input. Also, since APL has not been specified, info-area-1 will use the ASCII character set.

3.1.3 Defining A Viewport -

```
DEFINE VIEWPORT <viewport name>  
  DISPLAY AREA <w> <h>  
  FIX CORNER AT <x> <y>  
  {BORDER, NOBORDER}  
  END
```

This command establishes a portion of the terminal screen as a viewport with the name indicated, and with the operating characteristics specified, where

w specifies the width of the viewport in characters, and may have any value from 1 to the width in characters of the terminal screen (normally 80).

h specifies the height in lines of the viewport and may have any value from 1 to the number of lines on the terminal screen (normally 24).

x and y specify the x- and y-coordinates of the upper left hand corner of the viewport on the terminal screen. x represents a character position, counting from left to right; y represents a line number, counting from top to bottom. Thus the upper left hand corner of the terminal screen (the origin) is reckoned as 1 1.

If BORDER is specified, a single line will appear across the top of the viewport, with the viewport name centered along it, and a broken line will appear down the right-hand edge of the viewport. These borders occupy space allocated to the viewport, thereby decreasing the area available for displaying data.

For example, the set of commands:

```
DEFINE VIEWPORT viewport-1  
  DISPLAY AREA 40 10  
  FIX CORNER AT 1 6  
  BORDER  
  END
```

establishes an area of the terminal screen which is 40 characters wide and 10 lines high as "viewport-1", whose upper left-hand corner lies at the left most character position on the screen, on the sixth line from the top, and which has a border. In other words, viewport-1 is the left half of the terminal screen, from line 6 through line 15, with the border occupying the left half of line 6, and character position 40 of lines 6 through 15.

Having created each of the three basic WINDOW components, it remains only to link them together via the two CONNECT commands, and you will have established a sub-terminal.

3.1.4 Connecting A Channel To An Info Area -

```
CONNECT CHANNEL <channel name> TO <info area name>
```

This command logically connects the info area and channel indicated so that a program running under the control of the channel will communicate with the info area in the same way as it would communicate with a terminal. Continuing the example, the command

```
CONNECT CHANNEL CHANNEL-1 TO INFO-AREA-1
```

establishes the same relationship between info-area-1 and any program running in channel-1 as ordinarily obtains between a program and a standard terminal.

3.1.5 Connecting An Info Area To A Viewport -

```
CONNECT INFORMATION <info area name>  
TO <viewport name>
```

This command establishes a logical link between the info area and viewport named, and causes the contents of the info area to appear in the viewport. Although the logical link established by this command cannot be broken, it is possible to remove the viewport from the screen, thereby making the screen area available for displaying the contents of a different info area through a different viewport. For information on how to do this, see the explanations of the DISPLAY and UNDISPLAY commands in section 3.3.

Since this CONNECT command includes an implicit DISPLAY command, users who ignore the Sub-terminal Configuration Guidelines should consult section 4 for additional information on the effects of the DISPLAY command.

3.2 Keyboard Control

Having used the DEFINE and CONNECT commands to construct the configuration of sub-terminals needed for the task at hand, you are now in a position to begin actually running a program through each sub-terminal. Since each sub-terminal is treated by the computer operating system as if it were a separate terminal, you must begin by logging in each sub-terminal. When you do so, the operating system will assign to each sub-terminal being logged in a different "job number", just as if it were an ordinary terminal. It should also be noted that each sub-terminal may be logged in under a different project and/or programmer number. In order to log in a sub-terminal, however, you must first be able to choose the info area into which your keyboard input will go. Before you can log a sub-terminal in, you must use the TRANSFER command to redirect keyboard input, which has been going into COMINFO, so that it goes into the info area of the sub-terminal to be logged in.

3.2.1 Transferring Into An Info Area -

TRANSFER <info area name>

This command specifies the info area to which subsequent keyboard input will go.

Note: since all WINDOW commands must be entered into COMINFO, it is not possible to TRANSFER "directly" from one user info area to another. To redirect keyboard input from one user info area to another, first return to COMINFO by entering CONTROL-G as the first character of a new line, and then issue the appropriate TRANSFER command.

Thus, the command

TRANSFER INFO-AREA-1

causes subsequent keyboard input to be written into info-area-1, until the next CONTROL-G is entered, which directs keyboard input back to COMINFO.

3.3 Display Management

To facilitate the control of multiple simultaneously active programs, WINDOWS provides three display management commands: DISPLAY, CLEAR, and UNDISPLAY.

3.3.1 Displaying A Viewport -

DISPLAY <viewport name>

This command causes the contents of the info area CONNECTED to the viewport to be displayed in the area of the terminal screen assigned to it.

This command may be used to retype data which has been CLEARED from the screen, or to reinstate a viewport which has been UNDISPLAYed (see below). Thus, if viewport-1 has previously been CLEARED or UNDISPLAYed, its contents may be made to appear once more by entering the command

DISPLAY VIEWPORT-1.

Users who ignore the Sub-terminal Configuration Guidelines should consult section 4 for additional information on the DISPLAY command.

3.3.2 Clearing The Screen -

CLEAR {SCREEN, <viewport name>}

This command blanks out either the entire screen or the viewport named, but has no effect on the contents of the info area being displayed in the viewport. Thus, the command

CLEAR VIEWPORT-1

blanks out viewport-1.

3.3.3 Undisplaying A Viewport -

UNDISPLAY <viewport name>

This command causes the viewport named to cease reflecting changes in the info area to which it is connected. The data in the info area when this command is issued will continue to appear in the viewport even though new data is being written into the info area. Users who ignore the Sub-terminal Configuration Guidelines (section 3.1) should consult Additional Considerations Regarding WINDOW components (section 4).

3.4 Sub-terminal Maintenance Commands

Although the DEFINE commands are characteristically used at the beginning of a terminal session to establish the initial configuration of sub-terminals, they are by no means limited to such use. These commands may be used at any time either to set up new sub-terminals or to modify the characteristics of existing ones. To change an info area, use the DEFINE INFORMATION command, naming the info area to be changed, and including just those key words and variables that pertain to the operating characteristics you would like to change, followed by END.

Whenever you reference the name of a WINDOW component which has already been defined, WINDOWS will display a message noting that the name is already in use, and alerting you to the fact that it is modifying the characteristics of an existing info area or viewport (as the case may be) rather than creating a new one. When the dimensions or location of a viewport are modified, the information in the info area connected to the viewport being modified will be retyped. For example, to change the character set used in info-area-1 from ASCII (originally established by default) to APL, enter the command

```
DEFINE INFORMATION INFO-AREA-1
      APL
      END
```

The same procedure should be followed, using the DEFINE VIEWPORT command to change the characteristics of an existing viewport. Thus, the set of commands

```
DEFINE VIEWPORT VIEWPORT-1  
    DISPLAY AREA 80 10  
    NOBORDER  
END
```

increases the width of viewport-1 from the original 40 characters to 80 characters, and eliminates the border. The height of the viewport was initially 10 lines and is unaffected.

Two additional commands are provided to facilitate the reconfiguration of sub-terminals.

3.4.1 Obtaining The Status Of Sub-terminals -

```
STATUS {ALL, <info area name>, <viewport name>}
```

Displays in the command viewport the current specifications of info areas and/or viewports. Information provided includes all of the user-defined characteristics entered in the DEFINE INFORMATION and DEFINE VIEWPORT commands.

3.4.2 Disconnecting An Info Area From A Channel -

```
DISCONNECT <info area name>
```

This command breaks the logical link between the info area named and the channel to which it was most recently CONNECTED.

3.5 Miscellaneous Commands

Use of WINDOWS is facilitated by the PROFILE feature which makes it possible to initiate a frequently used series of WINDOW commands by entering a single command.

3.5.1 Calling A Profile -

PROFILE <file name>(<extension>)

Reads the file indicated and enters each record as a WINDOW command. If no extension is given, ".DPR" is assumed.

3.5.2 Exiting From WINDOWS -

When you have finished running all the programs you wish to run through WINDOWS, enter the command:

QUIT

into COMINFO. This command terminates processing of the Window Management System.

Note: before entering this command, the user should have logged off every channel ever logged in during the course of the WINDOW session. Any channel which has not been logged off when WINDOWS is terminated will remain logged in and will be "detached" by the operating system. This is a waste of computer resources and should be avoided. If for any reason WINDOWS is terminated or aborted before all of the channels are logged out, you should "attach" to each job number which was assigned to one of these channels and log it out. See installation user manuals for details on attaching and logging out.

4.0 ADDITIONAL CONSIDERATIONS REGARDING WINDOW COMPONENTS

Because the dimensions of info areas and viewports are specified separately, it is possible to display an info area in a viewport which is capable of displaying more or less data than is contained in the info area. If the viewport is larger than the info area, the contents of the unused screen area will be unpredictable. If, however, the viewport is smaller than the info area being displayed in it, the data in the info area is seen as if it were at the other end of a tunnel whose mouth is the size of the viewport and is fixed at the lower left hand corner of the info area.

Suppose, for example, that an info area which is defined as having 50-character lines is displayed through a viewport which is defined as 40 characters wide and 10 lines high. What appears on the screen in the area assigned to the viewport in question will be the left-most 39 characters of each of the ten lines most recently written into the info area. The right-most character position of each line in the viewport will contain a '>' to indicate that more information is available in the info area.

Although this kind of truncation might occur as the result of an oversight, it might also be intentional: one might choose to monitor the flow of "normal" data in an info area by viewing only part of each line, and then expand the viewport to display the entire contents of the info area when special conditions occur.

It is also possible to specify the dimensions and location of two (or more) viewports in such a way that all or part of one viewport is assigned to the same area of the terminal screen as all or part of another viewport. Since it is obviously impossible for an area of the physical screen to display two different sets of data at the same time, only one of any set of overlapping viewports is actually visible at a time, the rest being obscured by the one which is visible. In any such set of overlapping viewports, that viewport is currently visible which has most recently been referenced in a DISPLAY (or CONNECT) command (see 3.3.1 and 3.1.5).

Two points should be stressed regarding overlapping viewports: first, a viewport that is obscured is completely invisible, even if only part of it is overlapped by the viewport that is currently visible; also, the invisibility of a viewport has no effect on the flow of data in the info area connected to it. An important implication of these points is that data written into an info area (either by the user or by a program) may never actually appear on the

terminal screen. This would occur if the data are written into the info area when the viewport displaying the info area is obscured, and are on a line that is no longer being retained in the info area when the viewport becomes visible again. The only way to ensure that information will not be lost in this way is to observe the second window configuration guideline (3.1). By defining viewports so that none of them overlap, you can guarantee that all information written into each info area will appear on the screen (assuming that all of the viewports are always being displayed).

4.1 Implications For Specific Commands

4.1.1 DISPLAY <viewport-1> -

1. If viewport-1 is currently obscured by viewport-2, viewport-1 will become visible and obscure viewport-2.
2. If the info area to which it is connected is wider than viewport-1, each line of the info area is truncated at the right. When this occurs, a ">" is placed at the right-hand end of each viewport line to indicate that more data is available in the info area.
3. If the 30 lines retained in the info area are more than can be displayed at one time in the viewport, the viewport will show the most recent n lines of the info area, where n is the height of the viewport.

The above considerations also apply to the CONNECT command which links info areas to viewports, since this command includes an implicit DISPLAY.

4.1.2 UNDISPLAY <viewport-1> -

1. If viewport-1 was not obscuring any other viewport(s), the area of the terminal screen occupied by the viewport UNDISPLAYed will become frozen as described in section 3.3.3.

ADDITIONAL CONSIDERATIONS REGARDING WINDOW COMPONENTS

2. If viewport-1 was obscuring one or more other viewports, the most recently DISPLAYed of the other viewports will become visible (provided that it is not also obscured by another more recently DISPLAYed viewport).

If, for example, viewport-1 is currently visible and is obscuring viewport-2, the command

UNDISPLAY VIEWPORT-1

will cause the contents of viewport-1 to disappear from the screen, and make the contents of viewport-2 appear in the area assigned to viewport-2. Now viewport-2 is obscuring viewport-1. When the situation requires that viewport-1 be made visible again, this may be done by entering a DISPLAY command.

Summary of Window Commands

All WINDOW commands must be entered into COMINFO, the command info area. To return to COMINFO from another info area, enter CONTROL-G as the first character of a new line. COMMAND keywords are printed below in capital letters, and may be abbreviated as long as the abbreviation is unique. Lower case items which are enclosed in <corner brackets> are place-holders for user-supplied values. When a group of items is enclosed in {braces}, then one of the items must be chosen either explicitly by the user, or as a DEFAULT. All items in a group which is enclosed in (parentheses) are optional. User-supplied names of channels, info areas, and viewports must be unique, and may be any combination of up to 20 ASCII characters.

<u>Section</u>	<u>Command</u>
3.3.2	CLEAR {SCREEN, <viewport name>}
3.1.4	CONNECT CHANNEL <channel name> TO <info area name>
3.1.5	CONNECT INFORMATION <info area name> TO <viewport name>
3.1.1	DEFINE CHANNEL <channel name>
3.1.2	DEFINE INFORMATION <info area name> { <u>READWRITE</u> , READONLY} WIDTH <w> CHARACTERS {ASCII, APL} END
3.1.3	DEFINE VIEWPORT <viewport name> DISPLAY AREA <w> <h> FIX CORNER AT <x> <y> {BORDER, <u>NOBORDER</u> } END
3.4.2	DISCONNECT <info area name>
3.3.1	DISPLAY <viewport name>
3.5.1	PROFILE <file name>(<extension>)
3.5.2	QUIT
3.4.1	STATUS {ALL, <info area name>, <viewport name>}
3.2.1	TRANSFER <info area name>
3.3.3	UNDISPLAY <viewport name>

ODA DISTRIBUTION LIST

Director, Engineering Psychology
Programs (Code 455)
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Technical Information
Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

CDR Paul Chatelier
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

CAPT John Duncan
Office of the Secretary of Defense
(C3I)
Pentagon, Room 3C200
Washington, DC 20301

Dr. Craig Fields
Director, Cybernetics Technology
Office
Defense Advanced Research Projects
Agency
1400 Wilson Boulevard
Arlington, VA 22209

Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, DC 20350

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, DC 20380

Commanding Officer
ONR Eastern/Central Regional Office
ATTN: Dr. J. Lester
Bldg. 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. Charles Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Western Regional Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Tactical Development & Evaluation Support
Code 230
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Naval Analysis Programs
Code 431
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (2 cys)

Operations Research Program
Code 434
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Statistics and Probability Program
Code 436
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

ODA DISTRIBUTION LIST

Dr. James McGrath
Code 311
Navy Personnel Research and
Development Center
San Diego, CA 92152

Management Support Department
Code 210
Navy Personnel Research and
Development Center
San Diego, CA 92152

Naval Electronics Systems Command
Human Factors Engineering Branch
Code 4701
Washington, DC 20360

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, DC 20375 (6 cys)

Mr. Arnold Rubinstein
Naval Material Command
NAVMAT 08D22
Washington, DC 20360

Commander, Naval Electronics
Systems Command
Command and Control Division
Code 530
Washington, DC 20360

Dr. John Silva
Head, Human Factors Division
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 92940

Dr. Joseph Zeidner
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Donald A. Topmiller
Chief, Systems Effectiveness
Branch
Human Engineering Division
Wright Patterson AFB, OH 45433

Dr. H. W. Sinaiko
Smithsonian Institution
801 N. Pitt Street
Alexandria, VA 22314

Office of the Chief of Naval
Operations OP942
Pentagon
Washington, DC 20350

Office of the Chief of Naval
Operations OP987C
R&D Plans Division
Washington, DC 20350

ODA DISTRIBUTION LIST

Commander
Naval Electronics Systems Command
C3 Project Office
PME 108-1
Washington, DC 20360

CDR P. M. Curran
Human Factors Engineering Division
Code 604
Naval Air Development Center
Warminster, PA 18974

M. L. Metersky
Naval Air Development Center
Code 5424
Warminster, PA 18974

Dr. Edgar M. Johnson, Director
Organizations & Systems Research
Laboratory
U.S. Army Research Laboratory
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. David Dianich
Chairman, Dept. of Business and
Economics
Salisbury State College
Salisbury, MD 21801

Mr. Victor Monteleon
Naval Ocean Systems Center
San Diego, CA 92152

Commander, Naval Electronics
Systems Command
ELEX-03
Washington, DC 20360

CDR Richard Schlaff
NIPSSA
Hoffman Bldg. #1
2461 Eisenhower Avenue
Alexandria, VA 22331

Dr. Chantee Lewis
Management Department
Naval War College
Newport, RI 02840

Dr. John Shore
Naval Research Laboratory
Code 5403
Communications Sciences Division
Washington, DC 20375

Dr. Meredith Crawford
American Psychological Association
Office of Educational Affairs
1200 17th Street, NW.
Washington, DC 20036

Dr. William Dejka
ACCAT
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Merlin, Malehorn
Office of the Chief of Naval
Operations (Op 102)
Washington, DC 20350

Dr. S. D. Epstein
Analytics
2500 Maryland Road
Willow Grove, PA 19090

Dr. Amos Freedy
Perceptronic, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. G. Hurst
University of Pennsylvania
Wharton School
Philadelphia, PA 19174

ODA DISTRIBUTION LIST

Dr. Miley Merkhofer
Stanford Research Institute
Decision Analysis Group
Menlo Park, CA 94025

Mr. Tim Gilbert
MITRE Corporation
1820 Dolly Madison Blvd
McLean, VA 22102

Mr. George Pugh
Decision Science Applications, Inc.
1500 Wilson Boulevard
Arlington, VA 22209

Mr. Leslie Innes
Defence & Civil Institute of
Environmental Medicine
P. O. Box 2000
Downsview, Ontario M3M 3B9
Canada

Dr. Arthur Siegel
Applied Psychological Services
Science Center
404 E. Lancaster Street
Wayne, PA 19087

Dr. Rex Brown
Decision Science Consortium
Suite 421
7700 Leesburg Pike
Falls Church, VA 22043

Mr. David Walsh
Integrated Sciences Corporation
1640 Fifth Street
Santa Monica, CA 90401

Dr. A. C. Miller, III
Applied Decision Analysis
3000 Sand Hill Road
Menlo Park, CA 94025

LCDR J. A. Sears
Department of MIS
College of Business Administration
University of Arizona
Tucson, AZ 85721

I. R. Mirman
Asst for Special Projects
HQ AFSC-DL
Andrews AFB, MD 20334

Mr. Joseph Wohl
MITRE Corporation
Box 208
Bedford, MA 01730

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 OLN
ENGLAND

DA
FILM
O-