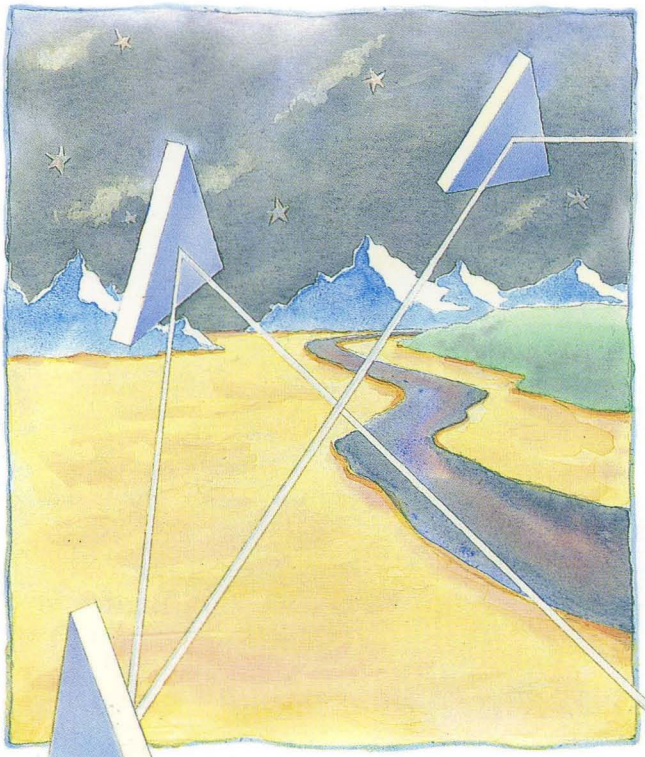


Domain/OS Display Manager Command Reference

011418-A00

apollo

Domain/OS Display Manager Command Reference

Order No. 011418-A00

Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824

First Printing: July 1988

Preface

The *Domain/OS Display Manager Command Reference* provides complete reference information on all the Display Manager commands that are available to you. We assume that you are already familiar with the material in *Getting Started with Domain/OS* and the User's Guide for your environment. Basics like file structure and usage are taken for granted here: this manual tells you how to use commands, not why you might want to use them.

We've divided the manual into two parts. Chapter 1 summarizes the basic concepts that apply to the Display Manager commands; Chapter 2 describes each command individually.

Documentation Conventions

This manual uses the following symbolic conventions:

commands and keywords	Bold words or characters in formats and command descriptions represent commands or keywords that you must use literally. Bold words in text indicate the first use of a new term. Filenames and pathnames are also in bold.
user-supplied values	Italic words or characters in formats and command descriptions represent values that you must supply.
example user input	In examples, information that the user enters appears in bold typeface.
`output`	Information that the system displays appears in `this typeface`.

[ ]	Square brackets enclose optional items in formats and command descriptions.
{ }	Braces enclose a list from which you must choose an item in formats and command descriptions.
|	A vertical bar separates items in a list of choices.

Related Manuals

The Help file **manuals** lists current revisions of all manuals for this software release.

Refer to the *Domain Documentation Quick Reference* (002685) and the *Domain Documentation Master Index* (011242) for a complete list of related documents. Refer to the following documents for more information on Domain®/OS, and the Aegis™, BSD, and SysV commands:

Getting Started with Domain/OS	(002348)
Aegis Command Reference	(002547)
BSD Command Reference	(005800)
SysV Command Reference	(005798)

Problems, Questions, and Suggestions

We appreciate comments from the people who use our system. To make it easy for you to communicate with us, we provide the Apollo® Product Reporting (APR) system for comments related to hardware, software, and documentation. By using this formal channel you make it easy for us to respond to your comments.

You can get more information about how to submit an APR by consulting the appropriate Command Reference manual for your environment (Aegis, BSD, or SysV). Refer to the **mkapr** shell command description. You can view the same description online by typing:

$ help mkapr (in the SysV environment)

% help mkapr (in the BSD environment)

$ help mkapr (in the Aegis environment)

Alternatively, you may use the Reader's Response form at the back of this manual to submit comments about the manual.

Contents

Chapter 1 Display Manager Basics

Chapter 2 Display Manager Commands

PERMUTED INDEX

push or pop a window on the stackwp ..... wp
move cursor right to next tab stopth ..... th
cursor left to previous tab stopthl move ..... thl
ts set tab stops for all windows ..... ts_dm
es insert string ..... es
first occurrence of matched stringso substitute ..... so
/all occurrences of matched string in defined range ..... s
matched string in defined/ s substitute all occurrences of ..... s
matched string so substitute first occurrence of ..... so
ds suspend a process ..... ds
dc continue a suspended process ..... dc
th move cursor right to next tab stop ..... th
move cursor left to previous tab stopthl ..... thl
ts set tab stops for all windows ..... ts_dm
in window tb move cursor to bottom line ..... tb_dm
window tdm move cursor to DM input ..... tdm
letters in a defined range of textcase change case of all ..... case
buffer xd cut (delete) text and write it to paste ..... xd
rubberbanding echo begin text echoing, end ..... echo
xp paste (write) buffered text into pad ..... xp
involving echo abrt abort text search; cancel any action ..... abrt
xc copy text to paste buffer xc
tab stop th move cursor right to next ..... th
previous tab stop thl move cursor left to ..... thl
window ti move cursor to next input ..... ti
beginning of the current line tl move cursor to the ..... tl
(previous) window tlw move cursor to last ..... tlw
tn move cursor to next window ..... tn
tni move cursor to next icon ..... tni
tt move cursor to top line in window ..... tt
pt move top of pad into window ..... pt
the current line tr move cursor to the end of ..... tr
pn save transcript pad in named file ..... pn
windows ts set tab stops for all ..... ts_dm
window tt move cursor to top line in ..... tt
specified window border twb move cursor to a ..... twb
kbd declare keyboard type ..... kbd
while maintaining edit pad unchanged/update edit file ..... pw
command(s) undo undo previous DM ..... undo
undo undo previous DM command(s) ..... undo
maintaining edit pad/ pw update edit file while ..... pw
wme move a window using rubberbanding ..... wme
set or display an environment variableenv ..... env
pv scroll pad vertically by lines ..... pv
pp scroll pad vertically by pages ..... pp
wi make a window or group visible or invisible ..... wi
wa set window autohold mode ..... wa
associated functions wc close window and ..... wc
positions wdf define DM default window ..... wdf
wg grow or shrink a window ..... wg
rubberbanding wge grow/shrink a window with ..... wge
window group wgra create or add to a ..... wgra
group wgrr remove window/group from ..... wgrr
wh set window hold mode ..... wh
visible or invisible wi make a window or group ..... wi
ap acknowledge alarm and pop window ..... ap

Chapter 1

Display Manager Basics

This chapter summarizes the basic concepts that apply to the Display Manager (DM) commands described individually in the following chapter. See the user guide for your environment for a detailed discussion of these concepts.

1.1 Defining Points and Regions

Unless otherwise noted, you must precede all DM commands by a pointing operation. This generally involves moving the cursor to the spot where the command is to be executed (for example, pointing to the window that you want to scroll), or specifying a specific screen or line location as a command argument. If you don't specify some pointing function, the DM executes the command at the current cursor position.

To point, simply move the cursor to the desired location. For example, to point to a window, place the cursor anywhere inside the window. The command reads the cursor position to determine which window you mean. Note that when you use the block cursor to designate a point on the screen, the designated point is at the lower left corner of the block cursor.

You can also define a point in any of the following ways:

line-number

Line numbers begin at 1 and range upward to the last line in the pad. Pads can contain up to 262,143 lines. You can use the symbol "$" to refer to the last line in the pad. Remember that the edit pad window legend contains the line number of the top line in the window for reference. You can also display the line number (plus the column number and x and y coordinates) of the current cursor position by using the DM command "=".

+/- *line-number*

The +/- *line-number* format denotes the *n*th line before or after the current cursor position in a pad.

[[*line-number*], [*column-number*]]

This format indicates the point by line and column number in the pad. The DM assumes the current line if you omit the first portion, and column one if you omit the second portion. Line numbers range from 1 to the last line in the pad (262,143 max.). Column numbers range from 1 to 256. When you specify a point in this format, you must use the outer set of square brackets to enclose the numbers. This is how the DM distinguishes between line/column positions in a pad and x/y coordinates on the screen (below). Note that the use of "$" to denote the last line in the pad does not work within square brackets.

Examples:

```
[127,14]     Line 127, column 14

[53]         Line 53, column 1.
             Brackets are optional
             in this case: see above

[,12]        Column 12 of the current
             line
```

([*x-coordinate*], [*y-coordinate*])

Screen coordinates specify bit positions on the display. The origin (0,0) is at the extreme upper left corner of the screen. Maximum values for x and y coordinates depend on the size of the screen in use. The DM uses the current x or y coordinate of the cursor if you omit it from the coordinate pair. When you specify a point in this format, you must use the outer set of parentheses to enclose the numbers. This is how the DM distinguishes between line and column positions in a pad (above) and x and y coordinates on the screen.

Examples:

```
(200,450)   Bit position with the given
            coordinates

(135)       Bit position whose x coordinate
            is 135 and whose y coordinate
            is the same as the current cursor
            position

(,730)      Bit position whose x coordinate
            is the same as the current cursor
            position and whose y coordinate
            is 730
```

/regular-expression/ or *\regular-expression* A regular expression specifies a string in the pad that begins or ends the region of interest. Regular expressions are described in Section 1.4.

Now that we can identify points, let's turn to regions. A **region** is simply the area between two points. Use the DM command **dr** (define region) to define a region. The region definition operation has the following format:

[*point*] **dr;** [*point*]

The first point marks one corner of the region; the second the opposite corner. Remember that you can use cursor positions to define the points or specify them explicitly in one of the alternate formats mentioned above.

For convenience, the predefined key, MARK, invokes the **dr** command. Point the cursor at the start of the range, mark it, then point at the end of the range.

When you use a DM command that requires you to specify a region to operate in, you can declare it either by marking it, or by explicitly specifying the region with one of the techniques described above. If a DM command does require you to define a region, specify the command in the following format:

[*region*]*command*

The symbol [*region*] indicates where you must define the region. Defining a range for text editing operations—cut, paste, substitute, and so on—is slightly different. See Section 1.3 for more information on defining text ranges.

1.2 Defining Window Boundaries

When a window's size or position on the screen is changed in any way, the DM determines the new boundaries of the window using calculations based on a pair of points (a "point pair") on the screen. Usually, the first point in the pair has been defined with the **dr** command and the second point is the current cursor position, although you can provide absolute point coordinates as described in Section 1.1.

Each point can specify either a new or an existing edge of a window, or a new or an existing corner of a window. The new window, then, is created based on the relationship between the x and y coordinates of the two points. When either point specifies a new upper edge or right edge for a window, the position is adjusted to account for the size of the displayed block cursor because the actual coordinates of the cursor are determined by its lower left corner. The DM makes this adjustment only when the coordinate source is the block cursor, not when the point comes from the touchpad or mouse, or from coordinates you enter explicitly.

The relationship between the two points in the point pair affects the actions of the window-related commands **cc**, **ce**, **cp**, **cv**, **wdf**, **wg**, and **wm** in the following ways:

1. Horizontal movement only (y coordinates of the two points are equal):

   Creation - Create a window bounded by the given x coordinates, the top of the screen, and just above the normal DM command window (that is, a full vertical window).

   wg/wm - Select the unobscured vertical edge nearest to the first point, and change the x coordinate of that edge to be that of the second point. The y coordinate of the first point must be within the unobscured range of y coordinates of the selected edge.

2. Vertical movement only (x coordinates of the two points are equal):

   This is analogous to horizontal movement, except that when creating, the DM uses the full horizontal width of the screen.

3. No movement (two points are equal):

   Creation - create a 512 by 512 window centered as nearly as possible (subject to screen boundaries) on the given cursor position.

   wg - treated as in Example 4 below.

   wm - Select the unobscured corner nearest the given point, and move the corner to that point.

4. Two points differ in both x and y:

   Creation - The given four coordinate values form opposing corners of the window.

   wg/wm - The first point selects the nearest unobscured corner (the corner itself must be visible) and that corner is repositioned at the second point.

If you specify only one point, that is, if you do not issue the **dr** command, grow is illegal and move behaves as in Example 3 above. The DM uses one of its five default window regions, or a default determined by the last window creation or deletion (**wc**) command, as follows:

- If the last such command was window deletion (that is, **wc**), the default region is the same as that of the deleted window.
- If the last such command was a successful window creation command, the default region is the next third of the screen following the created window.
- If the last such command was an unsuccessful window creation command, the default region is the same as specified in that command.

Use the DM command **wdf** to define the five default window regions.

1.3 Defining a Range of Text

The text editing commands that perform cut, paste, and substitute functions operate on a range of text. Declare that range just as you would mark any other region in a pad; that is, place the cursor at the start of the range, press <MARK>, then move the cursor to the end of the range and issue the command you want.

The region of text you define for a cut, paste, or search operation is highlighted in reverse video when you use <MARK>. This is because <MARK> invokes the **dr;echo** command sequence. You can still use the **dr** command alone to place a mark, but the highlighting feature is not invoked without **echo**. You can cancel the defined ranged with the **abrt** command. See the descriptions of the **dr**, **echo**, and **abrt** commands for more information.

Please note that the character under the cursor at the end of the range is not included within the range. Note also that you may not declare a range explicitly as an argument to the editing commands, since those commands do not, in general, accept arguments. You must use <MARK> or the **dr** command sequence.

The default range is different for these editing operations, too. While the general DM default range is the current cursor position, cut, paste, and substitute commands apply to all characters from the current cursor position up to the end of the

line (including the newline character) if you did not mark another range immediately before invoking the command.

1.4 Using Regular Expressions

Special regular expression notation is used to specify patterns for search and substitute strings in the DM editor. This notation is also used in the shell commands **ed** (edit), **edstr** (edit stream), **fpat** (find pattern), **fpatb** (find pattern block), and **chpat** (change pattern). Regular expressions permit you to describe textual patterns concisely without necessarily knowing their exact contents or format. You can create expressions to describe patterns in particular positions on a line: patterns that always contain certain characters and sometimes include others, or patterns that match text of indefinite length.

Regular expressions are constructed as follows:

1. Any standard ASCII character (except those discussed below) is a regular expression and matches one and only one occurrence of that character. (For multiple occurrence matches, see "*" below.) The case of the characters in the expression is not significant by default. Use the DM command sc (set case) to control case significance.

```
SAM
fred12          All valid expressions.
Joe(a&b)
```

2. Use a percent sign (%) at the beginning of a regular expression to match the empty string at the beginning of a line. If you put the % anywhere except at the beginning of the expression, it simply matches the percent character. Use this special feature to mark the start of a line in a regular expression.

   In the following example, **%Print** matches the string in line a but not in line b, because in line b **Print** is not at the beginning of the line.

```
(a)  Print this file.
(b)  This Print file.
```

3. A dollar sign ($) at the end of a regular expression matches the null character at the end of a line. If you put the $ anywhere except at the end of the expression, it simply matches the dollar sign character. Use this special feature to mark the end of a line in a regular expression.

   In the following example, file$ matches the string in line a but not in line b, because in line b file is not followed by an end-of-line marker.

```
(a)  Print this file
(b)  This file is permanent
```

4. A question mark (?) matches any single character except a newline character, unless you put the ? inside a character class (see below), in which case it represents the question mark character itself.

   ?OLD??? matches a and b, but not c, because in line c the letters **OLD** are alone on the line.

```
(a)  HOLDING
(b)  FOLDERS
(c)  OLD
```

5. While the ? matches only a single occurrence of a pattern, an asterisk (*) following a regular expression causes it to match zero or more occurrences of that expression, unless you put the * inside a character class (see below), in which case it represents the asterisk character itself. Matching zero or more occurrences of some pattern is called a closure. An expression used in a closure never matches newline.

   a*b Match **b**, **ab**, **aab**, and so on.

%a?*b	Match any string that begins with **a** and ends with **b**, and that is also the first string in the line. Any number of other characters can come between **a** and **b**.
[A-Z][A-Z][A-Z]*	Match any uppercase word; that is, any string containing at least two (and possibly more) uppercase characters (see Example 7, below). This expression does not match a string like **Mary** because **Mary** does not have two uppercase characters.

6. A string of characters enclosed in square brackets "[*string*]" is called a character class. This pattern matches any one character in the string but no others. However, if the first character of the string is a tilde (~), the regular expression matches any one character except the characters in the string. If you put the ~ anywhere except at the beginning of the string, it simply matches the tilde character. Note that the other special characters: %, $, ?, and *, lose their special meaning inside square brackets, and simply represent themselves.

[sam]	Match the single characters **s**, **a**, or **m**. If you want to match the word **sam**, don't use the square brackets.
[~sam]	Match any single character except **s**, **a**, or **m**.

7. Within a character class, you can specify any of a range of letters or digits by indicating the beginning and ending characters in the range, separated by a hyphen. That is, **0** through **9** matches any single digit; **a** through **z** or **A** through **Z** matches any single letter, lowercase or uppercase respectively. Remember, though, that the actual matching search ignores case unless you have used the DM command **sc** to enable case sensitivity. The range can be a subset of the digits or letters (that is, a through n or 3 through 8). However, the first and last characters of the range must be of the same type: digit, lowercase letter, or uppercase letter. "[A-9]" is illegal.

   Note that the "-" character has a special meaning inside square brackets. If you want to include the literal hyphen character in the class for matching, you must make it either the first or the last character in the class (so that it does not appear to separate two range-marking characters) or you must escape it (see Example 8, below).

   The "]" character is also special to character classes—it closes the class descriptor list. If you want to include the right bracket character in the class, you must escape it

In summary, the following characters have special meaning inside square brackets: ^ -]

[a-d]	Match any single occurrence of **a**, **b**, **c**, or **d**.
%[A-Z]	Match any capital letter that is also the first character on the line (%).
1-[1-9][0-9]*	Match any of the page numbers in this chapter.
[0A-Z]	Match any string containing a zero or a capital letter.
[^a-z0-9]	Match any uppercase letter or punctuation mark (that is, no lowercase letter or number).

8. The at sign (@) is an escape character. Characters preceded by @ have special meaning in regular expressions, as indicated below.

@n	Match newline character.
@t	Match a tab character. Note, however, that the keyboard TAB key does not insert a literal tab; instead, it moves the cursor to the display's next tab position. In a regular expression, **@t** matches only tab characters that were inserted with **@t**.
@f	Match a form feed character.

Use the escape character inside a character-class definition ([]) to specify literal occurrences of characters like "-" and "]" that have special functions inside square brackets. You can also use it whenever you need a literal occurrence of some special character in a normal expression (like ?, *, or @ itself).

[A-Z@-@]]	Match any capital letter, a hyphen, or a right bracket.
@?@*	Match a question mark followed by an asterisk, rather than zero or more occurrences of any character (?*).

9. You can concatenate regular expressions to form a more complex regular expression. The resulting regular expression matches the concatenation of the strings that the component regular expressions match. All the examples above concatenate expressions (single characters of some sort) into longer

strings for matching.

10. You can tag parts of a regular expression to help rearrange pieces of a matched string. A text pattern surrounded by braces "{pattern}" is remembered and can be referred to by @*n*, where *n* is a single digit referring to the string remembered by the *n*th pair of braces.

    s/{???}{?*}/@2@1/ s is the DM command for string substitution. The example moves a three-character sequence from the beginning of a line to the end of the line. "???" matches the first three characters of the line, and "?*" matches the rest of the line.

    so/{?}{?}/@2@1/ so is also a DM command for string substitution, but it substitutes only the first occurrence of the first pattern on a line. The example transposes two characters beginning with the one under the cursor. This is a handy key definition if you often type "ei" for "ie", and so on.

1.4.1 Summary of Features

c	Literal character
%	Beginning of line (if first character only)
$	End of line (if last character only)
?	Any single character except newline
*	Closure (zero or more occurrences of previous pattern)
[...]	Character class (any one of these characters)
[^...]	Negated character class (all characters except those in brackets or newline)
[*c1-c2*]	Any one of a range of characters from *c1* through *c2* (must be same type)
@c	Escaped character (for example, @@%, @@[, @@*, and so on)
{*expr*}	Tagged expression for use later in command line

Remember that the special characters described above apply only to regular expression operations. Some of these characters also have meanings (often radically different) in shell commands and other software products. If you are using a regular expression as part of one of those shell commands or products, be sure to enclose the expression in quotation marks so that the DM does not misinterpret it.

1.5 Key Naming Conventions

Every key on your keyboard (and mouse) has a name; in fact, almost every key has a set of three or four names. One set is the normal one, and is invoked when you press the key. The second set is invoked when you release the key; these are the up-transition names. The third set is invoked when you press the key simultaneously with the SHIFT key; these are the shifted names. Finally, many keys have special functions when you press them simultaneously with the CTRL key; these are the control shifted names.

1.5.1 Standard Key Names

The definable keys (see Figure 1-1) have the following names:

Letters and numbers
: These are named by their own single character. The capital letters are distinct from the lowercase letters: just refer to A instead of worrying about "a shifted". Enclose these keys in single quotation marks when you refer to them in a key definition.

ASCII Control
: These are the standard intraline and interline control keys.

CR	: Carriage Return
BS	: Backspace
TAB	: Tab
TABS	: Shifted Tab
CTRL/TAB	: Control Shifted Tab
ESC	: Escape; this is the same as 'CTRL/[' (hex 1B)
DEL	: Delete; this is the same as 'CTRL/\' (hex 7F)

Alphabetic Control
: These are named CTRL/x, where x is some other valid key name, for example, CTRL/Y or CTRL/N. There are also six non-alphabetic control characters; you must enclose their names in single quotation marks. The names and the hexadecimal values of the keys are: 'CTRL/[' (hex 1B), and 'CTRL/\' (hex 7F).

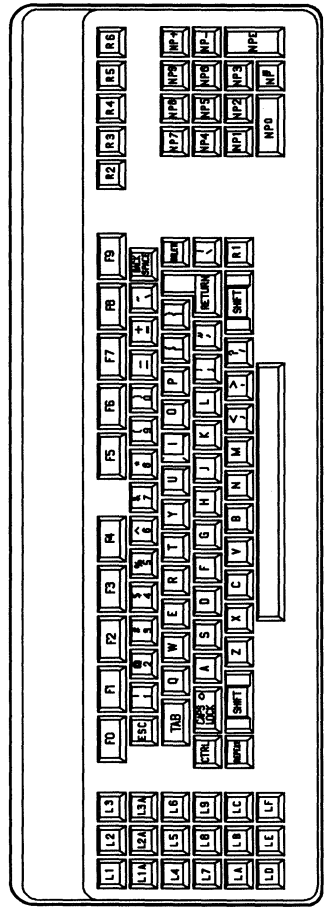


Figure 1-1. Keyboard Map

DM Function

These keys perform special DM functions. Those on the left side of the keyboard are named L1 through L9 and LA through LF. (Note that the low-profile keyboards have an extra row of keys below L1 through L3. These keys are named L1A, L2A, and L3A.) Their up-transition names are L1U through L9U and LAU though LFU. Their shifted names are L1S through L9S and LAS through LFS. The DM Function keys on the right side of the keyboard are named R1 through R6. Their up-transition and shifted names are formed in the same way that the left-side keys are.

Program Function

These keys are specially reserved for user program-control. They are at the top of the keyboard and are named F1 through F8, as labeled. Their up-transition names are F1U through F8U. Their shifted names are F1S through F8S. Their control shifted names are CTRL/F1 through CTRL/F8. (Note that the low-profile Model II keyboard has two additional program function keys, F0 and F9. Their shifted and control shifted names are derived as described above.)

Numeric Keypad

These keys are only available on the low-profile Model II keyboard. The keypad's numeric keys are named NP0 through NP9. The keypad symbols are named NP+, NP-, and NP. respectively. The ENTER key is named NPE. Keys 0 through 9, plus (+), and minus (-) can have shifted names (for example, NP+S).

Mouse

These are the keys located on the optional mouse pointing device (Figure 1-2). Their names are M1, M2, and M3. Their up-transition names are M1U, M2U, and M3U. There are no shifted or control shifted names.

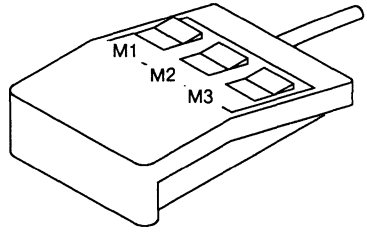


Figure 1-2. Mouse Key Map

You must enclose names containing special characters and all ordinary graphic characters in single quotation marks. For example, use the following command line to define the lowercase x key so that it acts just like the uppercase X key:

kd 'x' es 'X' ke

Although you can change the definitions of alphanumeric and special character keys, that capability is intended mainly for use in programs. When a program defines a key, the definition applies only while the program is running and only in pads the program controls.

1.5.2 Controlling Keys from Within a Program

Because of the great flexibility provided by our displays and keyboards, many applications programs assume control of these and redefine various capabilities. When this happens, the applications program overrides the default DM key definitions. The default definitions are restored once the applications program ends. For your own applications, you can control key definitions through program calls to the pad_$def_pfk and pad_$dm_cmd routines as described in the *Domain/OS Calls Reference*, Volume 1.

Because the normal functions of the DM keys are often useful (even when applications programs have redefined them), the HOLD and HOLD/GO keys are defined to provide a temporary override function. Pressing HOLD while in an applications program restores the keyboard to its log-in DM definitions. Pressing HOLD again re-enables the application-defined keys.

You cannot change this feature of the HOLD and HOLD/GO keys, which is functional only when the keyboard is under applications program control. This capability is independent of the default DM definitions of **wh** (window hold).

1.6 Special Characters in DM Scripts and Key Definitions

Several rules governing the use of literal and special characters affect the proper interpretation of commands within the DM environment. The following characters have special meanings when they appear in a DM command line or script.

@ The escape character "@" always nullifies any special meaning that the following character might have. As a part of command parsing, the DM strips off the "@" character itself. If you can't remember whether a character has some special meaning to the DM, it is always safe to escape the character—if it is not special, the DM still removes the "@", so the character appears as it should. The need for character escaping is generally confined to search and substitute operations, commands requiring quoted strings, and key definitions.

The use of "@" can be confusing in key definitions because the text in key definitions is processed twice: once when the definition is made, and again when the key is pressed and the definition is used. If you need to escape a character both times, you must precede it by three "@" signs. For example, "@@@#" becomes "@#" in the key definition, which then becomes "#" when the definition is used. Only the characters listed in this section are special within key definitions.

When read from a DM script, (via the **cmdf** command), the "#" character causes the remainder of the line to be treated as a comment and skipped.

; The semicolon is the normal command delimiter. It is equivalent to newline (generated by <RETURN>).

& The ampersand makes an input request, except when it is read from the keyboard. When read from the keyboard, it can be used in the replacement part of a substitution command to represent the entire string matching the regular expression. When "&" is preceded by "@" it becomes an ordinary character in both contexts. Therefore, you cannot use "&" within a script or key definition and also use its special meaning within substitute commands that appear in that script or definition.

Some commands accept strings surrounded by single quotation marks. They are **cp**, **cpo**, **cps**, **es**, **kd**, and the "&" character. When you use single quotation marks, the only characters in the quoted string that retain their special meanings are "@", "&", and the closing single quotation mark. All other characters revert to their literal graphic values. Note, however, that the **kd** command is not aware of single quotation marks within the definition string, so you must quote "#" and ";" there as well.

For example, to define the F4 key to enter the string "-#-" at the current cursor position, place the following line in a key definition file:

```
kd F4 es '-@@@#-' ke
```

Chapter 2

Display Manager Commands

NAME

aa – acknowledge display manager alarms

SYNOPSIS

aa

DESCRIPTION

The **aa** command acknowledges a Display Manager (DM) alarm. This command turns off the current alarm and enables further alarms, which may already be waiting. **aa** requires no arguments or options.

NAME

abrt – abort text search; cancel any action involving echo

SYNOPSIS

abrt

DESCRIPTION

The **abrt** command aborts a text search, and cancels any action involving the **echo** command.

When you use **abrt** to abort the current search, the DM returns the message "Search aborted." It does not move the window. Note that you must use a key already defined as "abrt" if you want to abort a search; typing **abrt** in the DM command-input window during a search does not work.

When you use **abrt** to abort the **echo** command, **abrt** cancels a move window with rubberbanding or grow window with rubberbanding operation; or it cancels highlighting for a defined range of text, depending on how you use **echo**.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other pad editing commands

NAME

ad – move cursor down one line

SYNOPSIS

ad

DESCRIPTION

The **ad** command moves the cursor down one line from its current position. By default, the down arrow key (LE) on the left-hand key pad executes this command.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys For other key definitions

NAME

al – move cursor left one character

SYNOPSIS

al

DESCRIPTION

The **al** command moves the cursor left one character from its current position. By default, the left arrow key (LA) on the left-hand key pad executes this command.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys For other key definitions

NAME

ap – acknowledge alarm and pop window

SYNOPSIS

ap

DESCRIPTION

When output is written to an obscured window, the DM signals you by beeping and displaying two bells in its alarm window. The **ap** command acknowledges the alarm and displays ("pops") the window to which the alarm pertains. This command is particularly useful if the window is completely covered so that you cannot point to it.

ap requires no arguments or options.

NAME

ar – move cursor right one character

SYNOPSIS

ar

DESCRIPTION

The **ar** command moves the cursor right one character from its current position. By default, the right arrow key (LC) on the left-hand key pad executes this command.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys For other key definitions

NAME

as – set scale factors for arrow keys

SYNOPSIS

as [*x* [*y*]]

DESCRIPTION

The as command sets scale factors for the arrow keys. The scale factor is useful for changing the apparent sensitivity of the arrow keys and for lining up the edges of windows after moving them.

ARGUMENTS

If you do not specify arguments, the default scale factors are used as described below.

x (optional) Specify a horizontal scale factor in raster units (integer). This value must be in the range 0-1023. (Note, however, that portrait displays may display only up to 800 raster units in this dimension.) There are approximately 100 raster units per inch. The default horizontal movement is the width of the character on which the cursor rests; if the cursor is not on a character, the DM uses the width of a space in the last window. Specifying 0 for *x* indicates that the default should be used.

Default if omitted: 0

y (optional) Specify a vertical scale factor in raster units (integer). This value must be in the range 0-1023. (Note, however, that landscape displays may display only up to 800 raster units in this dimension.) The default vertical movement is the height of a line in the last window. Specifying 0 for *y* indicates that the default should be used.

Default if omitted: leave current *y* value unchanged

NAME

au – move cursor up one line

SYNOPSIS

au

DESCRIPTION

The **au** command moves the cursor up one line from its current position. By default, the up arrow key (18) on the left hand key pad executes this command.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys For other key definitions

NAME

bgc – set background color of display

SYNOPSIS

bgc [-on | -off]

DESCRIPTION

The **bgc** command sets the background color for monochrome displays. Note that this is the display background only; the **inv** (invert_color) command controls window background color.

The background color is on, by default, at login.

OPTIONS

If you do not specify an option, **bgc** toggles the current mode.

-on Set the background color to grey or green, depending on display type.

-off Set the background color to black.

NOTE

bgc has meaning only for monochromatic displays. It has no effect on nodes with color displays. See the DM command **mono** for information about background color on color displays.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help inv	For details on setting window colors on monochrome monitors
help mono	For details on controlling window color on color monitors

NAME

bl – balance delimiters

SYNOPSIS

[*range*] **bl** [**-i** | **-c**] [*l_char*] [*r_char*]

DESCRIPTION

bl determines whether a given pair of delimiting characters (for instance, left and right parentheses) is balanced within a specified range of text.

ARGUMENTS

range (optional) Specify a range of text to be checked. This argument is valid only when used with **-c**; the range for **-i** is the current cursor position to the end (or beginning) of the file. Define the range to be checked as described in help DM range.

Default if omitted: check from cursor to end of line.

l_char (optional) Specify the left delimiting character. If you specify *r_char* but omit this argument, the left delimiting character defaults to *r_char*. If you omit both arguments, the left delimiting character defaults to left parenthesis.

r_char (optional) Specify the right delimiting character. If you specify *l_char* but omit this argument, the right delimiting character defaults to *l_char*. If you omit both arguments, the right delimiting character defaults to right parenthesis.

OPTIONS

If you specify either of the following options, it must precede any arguments you specify.

-i (default) Insert mode: search for balanced delimiters from the current character to the beginning or end of the file. The behavior of **bl** depends upon the character under the cursor when you invoke **bl**.

If you position the cursor on a delimiter and **bl** finds a balancing delimiter, it moves the cursor to the matched character momentarily (to show you where the balance is completed), then returns the cursor to the character immediately following the initial cursor postion. The search direction is forward if the character under the cursor is a left delimiter; backward if the character is a right delimiter.

If you position the cursor on a delimiter and **bl** finds no balancing delimiter, it gives an error message and sounds the alarm, then inserts a matching right delimiter at the initial cursor position.

If you position the cursor on a character other than a delimiter, **bl** searches backward for the first occurrence of *l_char*, briefly shows you where it is, then inserts a matching right delimiter at the initial cursor position.

-c Check mode: check only; do not insert balancing characters or move the cursor. You can mark a range of text to be checked if you specify this option; see the *range* argument above. **bl** checks all pairs of specified delimiters within the specified range and displays the results in the DM message window.

NAME

case – change case of all letters in a defined range of text

SYNOPSIS

[*range*] **case** [*options*]

DESCRIPTION

The **case** command changes the case of all the letters in a defined range of text. You can instruct **case** to invert the case of all letters, change all letters to uppercase, or change all letters to lowercase. If you do not specify a range, **case** operates on the text from the cursor position to the end of the current line.

OPTIONS

cc – create a copy of an existing window

SYNOPSIS

cc

DESCRIPTION

The **cc** command creates a copy of an existing window. With the cursor in the window to be copied, press <CMD> and issue the **cc** command.

If you do not mark a region for the new window with the **dr** command, **cc** uses the next default DM window to create the new window.

NOTES

There is a homonymous shell command: **cc** (compile_c) -- compile a C program.

cc does not work on GPR windows; if you use it on a GPR window, it displays a blank window.

NAME

cdm – change the display mode

SYNOPSIS

cdm [-p 1 | 8]

DESCRIPTION

The **cdm** command changes the display mode of the hardware that affects the colors the DM uses. You normally use this command in preparation for running a direct color application, which requires a 24-plane workstation. When you run such an application, you must restrict the DM to using only two colors.

At login, the default is **cdm** (with no options), which instructs the hardware to use the highest number of planes (normally 8) when drawing colors. This is an indirect color mode where the DM uses several colors for window banners, window background, and text.

Note that this command changes the colors on the screen of a 24-plane workstation only. It has no effect on any other display hardware and the DM gives an error message, "wrong display hardware", if you issue the **cdm** command on any device other than a 24-plane workstation.

The **cdm** command differs from the **mono** command in that the **mono** command does not affect the 24-plane hardware in any way. The **mono** command simply instructs the DM to use black and white for all its drawing operations, thus freeing up color slots in the color map.

OPTIONS

The only option that the **cdm** command takes is **-p** *n*, which allows you to specify the number of planes that the DM should use to get color. For example, **cdm -p 1** causes all DM output to be displayed in only two colors, through the use of one plane. This is necessary to free up all 24 planes so that some application can run in direct color mode. When you finish running a direct color application, you can restore the DM to its original state by issuing the **cdm -p 8** command. The default display mode for the DM is four colors for window background; four more for window banner background; white for banner text; and black for text in DM windows.

If you do not specify an option, **cdm** defaults to the highest number of planes, causing the display to be reset to its original state where existing indirect color applications work as before.

-p 1 This causes the DM to put the hardware in a state where the DM draws in only one plane, causing the DM's output to appear in two colors.

-p 8 (default) This instructs the DM to use all eight planes for drawing. It changes the hardware mode to allow the DM to use eight planes. The DM's output appears in many colors. This option is currently equivalent to giving the **cdm** command with no options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help scrattr	For information on how to list your screen characteristics
help mono	For information on ontrolling window color on color monitors
help inv	For information on controlling window color on monochrome monitors
help bgc	For information on controlling background color on monochrome monitors

NAME

ce – create an edit pad and window

SYNOPSIS

[*region*] **ce** *pathname* [*options*]

DESCRIPTION

Giving the **ce** command causes the DM to create an edit pad and a window in which to view it. If the file specified exists, the DM opens it for editing; if it does not exist, the DM creates and opens a file with the specified name.

By default, the EDIT key (r4) invokes the **ce** command, automatically moving the cursor to the DM input pad and issuing the "Edit file: " prompt. Type the pathname of the file to be edited.

Once an edit pad is created, you can use other DM commands to manipulate text in it.

Use the DM command **wc** to close a pad and window, without saving any changes you made. Use the DM command **pw;wc-?** to close a pad and window after saving any changes to disk.

ARGUMENTS

region (optional)	Specify the area of the screen where the new window will be displayed. Default if omitted: use the next DM default window
pathname (required)	Specify the file to be edited.

OPTIONS

-i	Specify that the window created for this pad should be in icon format, initially.
-c '*char*'	Specify the icon character, enclosed in single quotation marks, to be used in the icon window. *char* must reside in the current icon font. If you do not specify this option, and **-i** is present, the DM uses the default icon character for this pad type.

NOTE

The **ce** command does not create a process. It simply opens a file for editing within the current DM process.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help cv	For details on creating a read-only pad and window
help wc	For details on closing windows and pads

help dm commands	For a topical index of DM commands
help windows	For details on windows
help pw	For informtion on updating files

NAME

cmdf – execute DM script

SYNOPSIS

cmdf *pathname* [*option*]

DESCRIPTION

cmdf directs the DM to read commands from a file (DM script). When it reaches the end of the file, the cursor returns to its previous location.

You can nest command files; that is, you can use **cmdf** within another DM script.

ARGUMENTS

pathname (required) Specify the name of the file to be executed. You can specify DM commands one per line, or several per line, each delimited by semicolons.

OPTIONS

-p (optional) The **-p** option specifies that the command file to be executed is a paste buffer. The DM can execute commands in a paste buffer faster than it can execute the same commands in an ordinary file.

NAME

cms – erase existing marks

SYNOPSIS

cms

DESCRIPTION

The **cms** command erases any existing marks. Use it to ensure that commands requiring marked regions do not behave unexpectedly as a result of outstanding (but probably forgotten) marks. The LINE DEL standard key definition, **cms;tl;xd** is a good example: it clears previous marks and deletes only the current line.

cms requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dr	For details on placing marks
help gm	For details on locating marks

NAME

cp – create process, pads, and windows

SYNOPSIS

[*region*] **cp** [*options*] *pathname* [*args ...*]

ARGUMENTS

region (optional)	Specify the area of the screen where the new window will be displayed. Default if omitted: use next DM default window
pathname (required)	Specify the file to be executed by the new process: usually a shell (command interpreter).
args ... (optional)	Specify any arguments to be passed to the program *pathname*. If any of these arguments contains explicit blanks, enclose those arguments in quotation marks.

OPTIONS

Note that any options must precede the *pathname* argument.

-i	Specify that the window created for this process should be in icon format, initially.
-c '*char*'	Specify the icon character to be used in the icon window. *char* must reside in the current icon font. If you do not specify this option and **-i** is present, the DM uses the default icon character for this pad type.
-n *name*	Assign process name *name*. If you do not, the DM assigns the name "pad*n*," where *n* is an integer beginning with 1 and incremented by 1 for each active process.

EXAMPLES

1. Create a process named 'spare' running the shell. The **-nstart** option on **sh** suppresses startup file execution for the new shell.
   Command: **(0,0)dr;(500,300) cp -n spare/com/sh -nstart**

2. Create a process running the shell, and place it in a window in icon format using the default icon character for this pad type.
   Command: **cp -i /com/sh**

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help cpo	For details about creating processes without windows and pads
help cps	For details about creating server processes
help dq	For details about stopping processes

NAME

cpb – display a list of the windows in a group

SYNOPSIS

cpb *group_name* [*options*]

DESCRIPTION

The **cpb** command creates a window on a named paste buffer specific to the given group. The paste buffer contains a list of the windows in the group. Because these group lists are held in paste buffers, your programs can access the groups by using the pbufs routines described in the *Domain System Call Reference*.

The DM automatically creates three special paste buffers to help you manage your windows and icons. These paste buffers contain the following groups:

- The invis_group. This buffer holds the pathnames of all the windows that you have made invisible.

- The icon_group. This buffer holds the pathnames of all the windows that are represented by icons.

- The all_group This buffer holds the pathnames of every window open on your node: shell process windows, DM windows, visible and invisible windows, and windows represented by icons.

These special groups are created regardless of any other groups, and their members may overlap with the members of any other group (just as any group can have the same member(s) as another).

A special feature of the **cpb** command allows you to access the windows in a group directly, when the paste buffer holding the group is displayed on your screen. To use this feature do the following:

1. Use the **cpb** command to display the list of windows.
2. Position the cursor on the pathname of the window you want to access.
3. Press <CMD>, and issue the **dr** (mark) command.
4. Press <CMD> again, and issue the desired DM command.

By using this feature you can directly access windows that are invisible, represented by icons, and so on.

ARGUMENTS

group_name (required)	Specify the name of the group you want to display.

OPTIONS

-i	Specify that the window created will be in icon format.
-c '*char*'	Specify the icon character to be used in the icon window. *char* must reside in the current icon font. If you do not specify this option and **-i** is present, the DM uses the default icon character for this pad type.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wgra	For information on creating window groups
help icon	For information on creating icons

NAME

cpo – create process without pads or windows

SYNOPSIS

cpo [*options*] *pathname* [*args...*]

DESCRIPTION

The **cpo** command creates only a process, without associated pads or windows. The three standard I/O streams are directed to **/dev/null**. If this command appears in the node's DM boot startup script 'node_data/startup, the system assigns the new process the subject identifier (SID) user.server.none.local_node, and the created process continues to run regardless of whether anyone is logged in. This is desirable for utilities like the prsvr (print_server) and netman, and means that **cpo** is identical to **cps** in this context.

If **cpo** is issued in any other startup script or from the keyboard, the SID of the new process is derived from whatever process invokes **cpo**, and the created process terminates at logout.

ARGUMENTS

pathname (required)	Specify the file to be executed by the new process.
args... (optional)	Specify any arguments to be passed to the program *pathname*. If any of these arguments contain explicit blanks, enclose those arguments in quotation marks. Default if omitted: no arguments passed

OPTIONS

-n *name*	Assign process name *name*. If you omit this option, the process is not named.
-w	Invoke "wait" mode. If you specify this option, the DM suspends its activities until the newly created process terminates. As long as the process runs, the DM does not respond to keyboard or other input. Use this option with caution. If the newly created process does not terminate, the DM appears to be hung. In addition, processes created using **-w** cannot make any DM requests (via pad_$ requests or DM commands) because the DM is suspended and will not respond.

EXAMPLE

Run the alarm_server in a background process.

Command: **cpo /sys/alarm/alarm_server -disk 98 -bell**

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help cp	For details about creating processes with windows and pads
help cps	For details about creating server processes
help sigp	For details about stopping background processes

NAME

cps – create process independent of login

SYNOPSIS

cps [*options*] *pathname* [*args...*]

DESCRIPTION

cps creates a process (without associated pads or windows) that runs whether anyone is logged in or not. This is desirable for utilities like the prsvr (print_server) and netman. **cps** may appear in any of the DM startup scripts. However, you may prefer to issue the **cps** command from the keyboard on selected occasions, rather than include this function in a startup script.

The created process is assigned the subject identifier (SID) user.server.none.local_node regardless of the context in which the **cps** command appears. Be sure that any files to be used by this process (including the program specified by the *pathname* argument) give adequate access to this SID. If the access control lists (ACLs) on the files do not allow proper access to the server project name, the process terminates. Because background processes are essentially invisible, no error messages are returned to the display, making fault diagnosis difficult.

ARGUMENTS

pathname (required)	Specify file to be executed by the new process.
args... (optional)	Specify any arguments to be passed to the program *pathname*. If any of these arguments contain explicit blanks, enclose those arguments in quotation marks. Default if omitted: no arguments passed

OPTIONS

-n *name*	Assign process name *name*. If you omit this option, the process is not named.
-w	Invoke "wait" mode. If you specify this option, the DM suspends its activities until the newly created process terminates. As long as the process runs, the DM does not respond to keyboard or other input. Use this option with caution. If the newly created process does not terminate, the DM appears to be hung. In addition, processes created using **-w** cannot make any DM requests (via pad_$ requests or DM commands) because the DM is suspended and will not respond.

EXAMPLE

Run the server mbx_helper.

Command: **cps /sys/mbx/mbx_helper -n mbx_helper**

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help cp	For details about creating processes with windows and pads
help cpo	For details about creating non-server processes without windows or pads
help sigp	For details about stopping background processes
help servers	For details about available server programs

NAME

curs – control cursor positioning

SYNOPSIS

curs [-on | -off]

DESCRIPTION

curs controls whether a window is available for cursor positioning by the DM command **tn** (to_next_window), normally invoked by <NEXT WNDW>. All windows initially default to **curs -on**, which permits the DM to move the cursor into all windows via the **tn** command.

curs operates on a per-window, per-pane basis. For example, you may prevent the DM from moving the cursor to a transcript pad's pane while permitting it to move the cursor to the related input pad's pane.

To set the window state, simply point to the appropriate window and issue the **curs** command.

OPTIONS

If you do not specify an option, **curs** toggles the current mode.

-on Enable cursor positioning.

-off Disable cursor positioning.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tn For information on moving the cursor to various windows

NAME

cv – create a read-only edit pad and window

SYNOPSIS

[*region*] **cv** *pathname* [*options*]

DESCRIPTION

The **cv** command creates a read-only edit pad to view an existing file. You may not make changes to the file, only view it. If you decide that you want to make changes to it after all, you must first disable read-only mode. See the **ro** command description for details about that operation.

By default, the READ key (r3) invokes the **cv** command, automatically moving the cursor to the DM input pad and issuing the "Read file: " prompt. Type the pathname of the file to be read.

To close a pad and window, use the DM command **wc**.

ARGUMENTS

region (optional) Specify the area of the screen where the new window will be displayed.

Default if omitted: use next DM default window

pathname (required) Specify the file to be viewed. An error occurs if the file does not exist.

OPTIONS

-i Specify that the window created for this pad will be in icon format.

-c '*char*' Specify the icon character to be used in the icon window. *char* must reside in the current icon font. If you do not specify this option and **-i** is present, the DM uses the default icon character for this pad type.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ce	For details about editing writable pads
help wc	For details about closing windows and pads
help icon	For details about changing windows into icons
help dm commands	For a topical index of DM commands
help windows	For general information about windows

NAME

dc – continue a suspended process

SYNOPSIS

dc

DESCRIPTION

The **dc** command restarts a process that has been suspended by the **ds** (debug_suspend) command. Refer to the **ds** command description for details about that operation.

dc requires no arguments or options.

NAME

dq – generate a quit fault in a process

SYNOPSIS

dq [*entry_name*] [*options*]

DESCRIPTION

The **dq** command generates a quit fault, which normally interrupts execution of the current program and returns the process to the calling program. This command affects the process associated with the window that contains the cursor.

ARGUMENTS

entry_name (optional) Specify the name of the window or window group whose process is to receive the fault. Note that this is valid only for processes with windows. To stop background processes, use the shell command **sigp** (signal_process). If the name of the window or group appears as a text string somewhere on the display, you may use the following time-saving feature: place the cursor on the name, then press <MARK>. Now issue the **dq** command. **dq** uses the marked name for the *entry_name* argument.

Default if omitted: send fault to the process whose window is under the cursor

OPTIONS

If you do not specify an option **dq** generates a normal quit fault and halts whatever program is currently running.

-c *nn* Generate an arbitrary asynchronous fault with the specified hexadecimal status (*nn*).

-s Stop the entire process in a controlled way, if possible. Close open streams, files, pads, and so on. The shell's parent process is stopped and closed, too.

-b Blast process; do not execute further user-mode instructions. Open streams, files, and pads are not closed. If you blast processes, you should shut your node down and reboot.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ds For details about suspending a process

help dc For details about restarting a suspended process

help eef For details about stopping a shell and its process

NAME

dr – place a mark to define a region

SYNOPSIS

dr

DESCRIPTION

The **dr** command marks some part of the display or some part of a pad. You can use the mark to define a region for a substitute command, to grow, shrink, or move a window, or to reposition the cursor.

You can specify a literal point at which the mark is to be placed by preceding the **dr** command with line and column numbers in a pad, x and y screen coordinates, or regular expressions for matching text. If you do not specify a point, the mark is placed at the current cursor position.

By default, the MARK key invokes the **dr** command along with **echo** to provide user-visible feedback.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help gm	For details on locating marks
help cms	For details on erasing marks
help echo	For details on the user-visible feedback mechanism
help dm range	For a discussion of marking text ranges and window regions

NAME

ds – suspend a process

SYNOPSIS

ds

DESCRIPTION

The **ds** command generates a suspend signal for a process group. All activities are suspended. You can restart processes with the **dc** (debug_continue) command.

ds requires no arguments or options.

NAME

echo – begin text echoing, end rubberbanding

SYNOPSIS

echo [-r]

DESCRIPTION

When used as part of the **dr; echo** command sequence invoked with <MARK>, the **echo** command performs two separate operations depending on the situation. When you press <MARK> to begin defining a range of text, **echo** tells the DM to begin highlighting the indicated text range in reverse video (text echoing). When you use <MARK> to complete a move window (**wme**) or grow window (**wge**) operation, the **echo** command tells the DM to remove the "rubberband" and move or grow the window as indicated. Use the **sq** command to abort text highlighting or rubberbanding.

echo's main use is to highlight text that you mark for cutting, pasting, and copying with the **xd**, **xp**, and **xc** commands, respectively.

OPTIONS

-r Specify **echo** for a rectangular region of text. Use a mark point and the cursor to specify a column along the left side of the text you want to highlight in reverse video. When you issue the **echo** command with the **-r** option all text to the right of the specified column is diplayed in reverse video.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wge	For details on how to grow windows with rubberbanding
help wme	For details on how to move windows with rubberbanding
help windows	For a general description of how to move and grow windows, and how to define a range of text.
help xc	For information on copying text
help xp	For information on pasting text
help xd	For information on deleting text

NAME

ed – delete character under cursor

SYNOPSIS

ed

DESCRIPTION

The **ed** command deletes the character under the cursor. If the character is a newline, **ed** joins two lines. By default, <CHAR DEL> invokes the **ed** command.

ed requires no arguments or options.

NOTE

There is a homonymous shell command: **ed** -- invoke the line mode editor.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ee For details about deleting the character preceding the cursor

NAME

ee – delete character preceding cursor

SYNOPSIS

ee

DESCRIPTION

The ee command deletes the character preceding the cursor. If the window is in overstrike mode, ee replaces the preceding character with a blank. By default, the BACKSPACE key invokes the ee command.

ee requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ed_dm For details about deleting the character under the cursor

NAME

eef – insert end-of-file mark

SYNOPSIS

eef

DESCRIPTION

The **eef** command inserts a stream end-of-file mark (EOF) in the pad. If the line containing the cursor is empty, the EOF is written on that line. Otherwise, the EOF is inserted following the current line.

It is a common (although not universal) convention for programs to terminate execution and return to the process that called them when they receive an EOF on their standard input stream. The command shell is such a program. When the top-level program in a process returns, the process stops and all its streams are closed. The DM then closes the shell's process input pad and window, and closes the transcript pad. Whether the transcript window also disappears depends on the setting of its auto-close mode. If auto-close is disabled (the default condition), you must manually delete any windows associated with the closed transcript pad by using the DM command **wc -q**.

NAME

ei – set insert/overstrike mode

SYNOPSIS

ei [-on | -off]

DESCRIPTION

The **ei** command puts the current pad into (**-on**) or out of (**-off**) insert mode. If you do not supply an option, **ei** inverts the current mode. In insert mode, characters you type are inserted into the pad without replacing or overstriking any existing characters. This causes existing text to drift to the right as new text is added. In overstrike mode (that is, with insert mode turned off), characters you type at the keyboard replace those under the cursor. This can be useful for entering information into pre-formatted files so that the format is undisturbed.

By default, <INS> invokes the **ei** command without options to toggle the current mode.

The window legend contains an "I" when the window is in insert mode. The "I" disappears in overstrike mode.

All pads are initially in insert mode, although this is irrelevant if the pad is also read-only.

OPTIONS

If you do not specify an option, **ei** toggles the current mode.

-on Turn on insert mode.

-off Turn off insert mode.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ro	For details about setting pad read/write mode
help ws	For details about setting window scroll mode
help wa	For details about setting window autohold mode
help wh	For details about setting window hold mode

NAME

en – insert newline

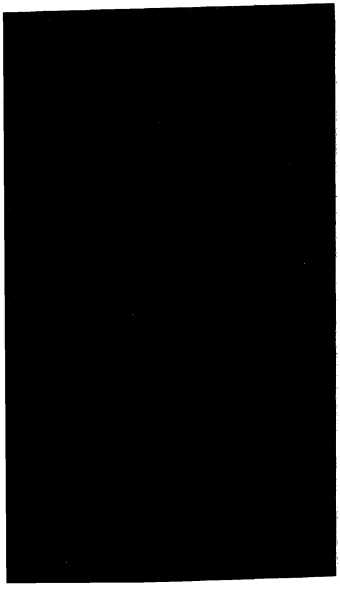
SYNOPSIS

en

DESCRIPTION

The **en** command inserts (or overstrikes, depending on current mode) a newline character at the current cursor position.

By default, the RETURN key invokes this command.



NAME

env – set or display an environment variable

SYNOPSIS

env *variable* [*value*]

DESCRIPTION

The DM command **env** sets or displays the value of an environment variable. Environment variables are of primary concern to Domain®/OS users; please consult the Domain/OS documentation for details about their usage.

If you invoke **env** from the keyboard, you may use it only to display environment variables, not set them. To set variables, **env** must appear in one of your startup scripts so that it gets executed before any shells are created, because the DM assigns values to environment variables for new shells using those in effect for the window that currently contains the cursor. **env** thus does not have a chance to influence the new shell if other shell(s) already exist. In addition, the **env** command never changes the value of a variable in an existing process.

ARGUMENTS

variable (required) Specify the name of the variable whose value is to be set or displayed. Since the DM normally forces arguments to uppercase prior to command scanning, enclose a variable whose name must be lowercase in single quotation marks.

value (optional) Specify the new value to be assigned to *variable*. Since the DM normally forces arguments to uppercase prior to command scanning, enclose a value that must be lowercase in single quotation marks.

Default if omitted: display the current value of *variable*

EXAMPLES

env systype Display the current value for systype for the current shell window.

env systype 'bsd4.3' Set the systype variable to 'bsd4.3'. This line must appear in a startup script to have any effect.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help export For information about manipulating environment variables from the shell. (**export** is an Aegis shell command.)

NAME

er – insert raw character

SYNOPSIS

er *nn*

DESCRIPTION

The **er** command sends a raw character to a program. The single argument *nn* (required) is a one-character or two-character hexadecimal value that defines the single byte sent to the active program the next time the program requests input. The data byte is not echoed anywhere on the display. In effect, this command delivers a single raw keystroke to a program.

Use the **er** command in programs that must define keys to return known values for actions by the programs.

This command differs from the other text insertion commands in that it does not insert the hexadecimal character into an edit pad. Its sole function is to pass a hexadecimal character to a running program.

NAME

es – insert string

SYNOPSIS

es *'string'*

DESCRIPTION

If a window is currently in write mode, then any text character typed at the keyboard is inserted at the current cursor position. This is the default DM action. Typing text into a read-only window causes an error.

The **es** command inserts a string of text at the current cursor position. Enclose the string to be inserted in single quotation marks. Since text insertion is the default action anyway, this command is primarily useful in key definition commands where you want some text written out when the key is pressed, or in DM scripts for writing text to the display.

NAME

ex – exit DM to boot shell

SYNOPSIS

ex

DESCRIPTION

The ex command causes the system to stop the DM process and enter the boot shell. This puts you in the same place that you would be if you had powered up your node with the normal/service switch set to service.

To restart the DM, type

) go

in the boot shell.

This command differs from **shut**, which shuts the node's operating system down completely and enters the Mnemonic Debugger that resides in the node's boot programmable read-only memory (PROM).

Do not confuse this command with the shell command **exit**, which exits a shell script loop.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help shut	For details about shutting down your node
help lo	For details about normal log out

NAME

fl – load a font for use in pads

SYNOPSIS

fl *pathname* [**-i**]

DESCRIPTION

The **fl** command loads a font for use in subsequent pads. Note that fonts apply to pads, not windows, so any new window opened to an old pad uses the old font.

You can load up to 50 fonts. The DM keeps track of fonts loaded by the **fl** command or programs. It unloads fonts on a least-recently-used basis.

If you need to unload a font (to edit it with **edfont** for example), issue an **fl** command for another font and close all the windows using the font you wish to unload. If a program loaded the font, stop the program, and close the window.

ARGUMENTS

pathname (required) Specify the name of file containing font to be loaded. The DM first looks up the given pathname directly, using the user working and naming directory rules. If it does not find the pathname, the DM then looks in the directory /sys/**dm**/**fonts**.

OPTIONS

-i Specify that the font to be loaded is an icon font.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help fonts For details about standard fonts that we supply

help icons For details about standard icons that we supply

help edfont For details about the character font editor

NAME

gm – go to a mark

SYNOPSIS

gm

DESCRIPTION

The **gm** command repositions the cursor at the most recently marked point after first marking the current cursor position (where you invoked the **gm** command). This allows you to alternate between two points with repeated invocations of **gm**. The most common use of this is to "remember" a position in a file and return to it later. **gm** repositions the window, if necessary, to display the marked pad location. **gm** requires no arguments or options.

NOTE

The mark stack is two deep, the most recently marked points replace the earlier marks. **gm** therefore allows toggling between two marked points.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dr	For details on placing marks
help rm	For details on restoring marks to the mark stack
help cms	For details on erasing marks

NAME

icon – change a window or window group into an icon(s); change an icon

SYNOPSIS

icon [*entry_name* [**-i**|**-w**] [**-c** *'char'*]]

DESCRIPTION

The **icon** command changes the specified window or group into an icon(s), or changes an icon back into a window. You can use two methods to change a window into an icon:

- Specify the window name (shown in the window legend) when you issue the **icon** command, (either by typing it or using <MARK> as described below).
- Simply position the cursor in the window, press <CMD>, and issue the **icon** command.

If you want to change a group of windows into icons, you must specify the group name when you issue the **icon** command. By default, if you do not specify an *entry_name* (the name of a window or group) **icon** manipulates the window under the cursor.

To change an icon back into a window, repeat the process described above. The window reappears on your display at its former position.

When you change a window into an icon, the DM displays an icon character that describes the type of information the window displayed, such as an edit pad, a graphics file, or a shell transcript pad. The default icon characters are held in a font file called **/sys/dm/fonts/icons**. If you want, you can use the edfont program described in the appendixes to examine or change this file. If you want to use your own icon font file, invoke the **fl** (font_load) command with the **-i** option prior to issuing the **icon** command.

ARGUMENTS

entry_name (optional) Specify the name of the window or group you want to change into icon(s), or change back into a window. If the name of the window or group appears as a text string somewhere on the display, you can use the following time-saving feature: place the cursor on the name, then press <MARK>. Now issue the **icon** command. **icon** uses the marked name for the *entry_name* argument.

Default if omitted: manipulate the window under the cursor

OPTIONS

If you do not specify any options, **icon** toggles the current window setting.

-i	Force the window or group to appear as an icon. This option is not valid if you specify **-w**.
-w	Force the window or group to appear as a window. This option is not valid if you specify **-i**.
-c '*char*'	Specify the DM icon character used to represent a window. You must enclose *char* in single quotation marks.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help icons	For a list of the standard icon characters residing in **/sys/dm/fonts/icons**
help fl	For information on loading icon font files for use in windows
help idf	For information on defining default icon positions

NAME

idf – set the icon default positioning and offset

SYNOPSIS

[*region*][*shift position*]; **idf**

DESCRIPTION

The **idf** command sets the position of an icon on your screen, determines where subsequent icons will be positioned (the offset), and specifies the icon shift vector to use when icons start to overlap each other. Each time you issue the **idf** command you reset the positions where any subsequent icons appear.

By default, icons appear in a horizontal line along the top of portrait displays, and in a vertical line along the right side of landscape displays. The default offset is set at the width of one icon (60 bits) horizontally or vertically, depending on the display. You can use **idf** to change this default positioning and offset, to establish the position of an icon created in a script, or to set your personal icon positioning and offset in a DM startup script (startup_dm). Specify the **idf** command in one of the following ways:

- Move the cursor to the new default icon position. Issue the **idf** command.

  This operation sets the first icon position; the offset of the next icon is 0,0 (pixels) and the shift vector is 0,0 (pixels). Therefore, all subsequent icons appear on top of one another at the first icon position.

- Move the cursor to the new default icon position. Use <MARK> or issue the **dr** command to mark the cursor position. Move the cursor to indicate the offset for the next icon. Issue the **idf** command.

  This operation sets the first icon position and next icon offset. The shift vector is 0,0 (pixels). Therefore, when icons need to use occupied positions, the DM places new icons directly on top of existing icons.

- Move the cursor to the new default icon position. Use <MARK> or issue the **dr** command to mark the cursor position. Move the cursor to indicate the offset for the next icon and once again issue a <MARK> or **dr** command. Then move the cursor again to set the shift vector for reused icon positions.

  This operation sets the first icon position and the next icon offset. It also establishes a shift vector so that icons do not appear directly on top of one another if the DM needs to place new icons over existing ones.

- Specify the icon position, offset, and shift explicitly in a command line. The format is as follows:

  (*first_xy_pos*)**dr**;(*next_xy_pos*)**dr**;(*shift_xy_pos*);**idf**

For example, the command line

Command: **(800,10)dr;(850,60)dr;(820,10);idf**

places the upper-left corner of the first icon at bit position (800,10), sets the icon offset vector to (50,50) (found by subtracting the 'initial' from the 'next' bit positions), and sets the shift vector to (20,0) (found by subtracting the 'initial' from the 'shift' bit positions). Therefore, the next icon appears at bit positions (850,60), the next at (900,110), and so on. If an icon must be placed on top of the first icon, it is positioned at (820,10), the next at (870,60), and so on.

The **idf** command requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help icon For more information on icons

NAME

inv – set window color

SYNOPSIS

inv [-on | -off]

DESCRIPTION

The **inv** command sets the color of all windows on monochrome displays. Note that these are window backgrounds only; **bgc** (background_color) controls the display background.

The window color is on, by default, at login.

OPTIONS

If you do not specify an option, **inv** toggles the current mode.

-on	Display black characters on a white or green background, depending on display type.
-off	Display white or green characters on a black background, depending on display type.

NOTE

inv has meaning only for monochromatic displays. It has no effect on nodes with color displays. See the DM command **mono** for information about window color on color displays.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help bgc	For details on setting the display background color on monochrome monitors
help mono	For details on controlling window color on color monitors

NAME

kbd – declare keyboard type

SYNOPSIS

kbd [{2|3}][-i]

DESCRIPTION

kbd allows you to specify the keyboard that is attached to your node so that the proper set of standard key definitions may be applied. When this command is invoked in the '**node_data_startup** file, it causes the DM to execute the corresponding key definition file (/sys/**dm**/**std_keys2** or /sys/**dm**/**std_keys3**).

If the '**node_data/startup** file does not invoke the **kbd** command, the DM does the following. It first tests for a low-profile Model II keyboard and if one is attached, it uses '**kbd** 3'. If a Model II keyboard is not present, the DM defaults to '**kbd** 2' (low-profile Model I keyboard).

ARGUMENTS

id (required)	Specify the keyboard ID. Valid IDs are 2 for the low-profile Model I keyboard, and 3 for the low-profile Model II keyboard.

OPTIONS

-i	Prompt for the type of keyboard attached to your system. The DM displays the type in its output window.

NOTE

kbd is valid only in the DM file '**node_data/startup**; you cannot type it from the keyboard. See the User's Guide for your environment for information on start-up files.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys	For a list of standard key definitions for both keyboards

NAME

kd – set or display key definition

SYNOPSIS

kd *key_name* [[*definition*] **ke**]

DESCRIPTION

The **kd** command defines a keyboard key as a sequence of DM commands. It also can display the definition of a key.

ARGUMENTS

key_name (required) Specify the name of the key to be defined or displayed. Key names are available from help DM keys. Enclose normal alphanumeric and punctuation keys in quotation marks.

definition (optional) Specify the sequence of DM commands that represent the desired key function; separate commands with newlines or semicolons. Definition can be any number of commands, but cannot exceed 1024 characters. Definitions may contain other predefined keys (that is, key definitions may be embedded in one another).

You must precede the input request character, '&', which is frequently used in key definitions, by an escape character when the **kd** command appears in a script.

If you do not specify a definition and **ke** is present (that is, the definition is null), the current key definition is deleted and the key reverts to its normal graphic value, if any. If **ke** is also absent, the definition of the named key is displayed in the DM message window.

Default if omitted: see above

ke (optional) Signal the end of the **kd** command. This argument is required if you specify a definition, or if you wish to delete a definition by specifying a null definition.

Default if omitted: display *key_name* definition

EXAMPLES

kd l3	Display definition of key 13.
kd f6 au;tr ke	Define f6 key to move the cursor to end of previous line in window.
kd ^C ke	Delete current definition of ^C.

You can embed key definitions in key definitions, and thereby define keys that define other keys. The embedded key definition follows the same rules as any other key definition. The **ke** that ends the embedded definition must be separated from the next command by an "escaped" semicolon; that is, a semicolon preceded by the @ character. For example,

kd kd ^X es 'April is the cruelest month' @ke; pv ke

changes the definition of the f3 key, which normally just invokes the DM command **pv**, so that it also changes the definition of ^X to print out the string shown. If you do not precede the ';' by an escape character, the DM does not accept the definition.

Note that key definitions within key definitions are scanned three times: 1) when the outer key definition is made, 2) when the outer key definition is executed and the inner key definition is made, and 3) when the inner key definition is executed. Therefore, be very careful when you escape (with "@") certain special characters such as "@" itself.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm keys	For a list of standard key definitions

NAME

l – log in to a node

SYNOPSIS

l *id.* [*group* [*org*]]

DESCRIPTION

The **l** command allows you to log in to a node. It is valid only at the beginning of a session in reponse to the "login:" prompt. Typing the **l** command after logging in causes an error. After you enter the **l** command, the system requests a password. If you specify either the ID or the password incorrectly, the system displays an error message and the correct format of the **l** command, and you can try again.

If you forget your password, you must contact your system administrator, who can assign a new password to you. The administrator cannot tell you your current password, because those are encrypted within the system and are not human-readable.

The 'l' character itself is optional when preceded by the "login:" prompt. You can omit it and simply type your ID if you want.

When you have logged in successfully, the system sets the working directory to your login home directory, which may be anywhere in your file hierarchy that you please. The system administrator for your network establishes the login home directory name when your account is created.

ARGUMENTS

You can separate the *id*, *group*, and *org* arguments either by periods (as shown), or by blanks.

id (required)	Specify the user ID assigned to you by the system administrator when your access privileges were established.
group (optional)	Specify the group ID associated with this user ID. User IDs may or may not have group IDs, depending on how the access privileges were set up.
org (optional)	Specify the organization ID associated with this user and group ID. Again, this may or may not be necessary for any particular user ID.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help login	For general information about logging in
help login window	For details about logging in to a running process
help lo	For details about logging off a node

NAME

lo – log out from a node

SYNOPSIS

lo [*options*]

DESCRIPTION

The **lo** command stops all user processes (except those created by the **cps** command and those created by the node startup file '**node_data/startup**), closes all pads and files, and returns the display to the "Please log in:" prompt.

If **lo** cannot terminate all active processes normally, the command asks you if you wish to blast the remaining processes (see **-f** below). Respond either "y" or "n".

You can also disable the ability to log out. See **-off** below.

You can execute a DM command script automatically at logout. The logout script must be in a file named '**node_data/startup_logout.***type*, where *type* is one of the standard display type extensions used for startup filenames (for example, 19l, color, none, and so on). Note that you cannot start up new processes with **cp**, **cps**, or **cpo** from this script, because the DM is in the process of shutting down all existing processes, unless you specify the **-w** option with **cps** or **cpd**.

OPTIONS

-f	Force logout by blasting processes that cannot be stopped normally. If you use this option, be aware that you may lose some disk space. Use the salvager **salvol** to recover the disk space. You may also lose files and programs that you had been working with. Therefore, use the **-f** option as a last resort when the normal logout procedure is not working.
-off	Disable the ability to log out. When this option is specified, the user who is currently logged in cannot log out.
-on	Enable logout. Use this option to restore a user's ability to log out.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help l	For details about logging in to a node
help ex	For details about exiting the DM to the boot shell without unloading Domain
help shut	For details about logging out of a node and shutting it down
help salvol	For details about using the disk salvaging utility

NAME

mono – set color monitor to black and white

SYNOPSIS

mono [-on|-off]

DESCRIPTION

mono controls whether the DM displays text and windows in color or in black and white. This command operates on color displays only. For information on controlling window color on monochrome displays, see the DM commands **bgc** (background color) and **inv** (invert color).

mono is off, by default, at login.

NOTE If you have enabled monochrome mode (**mono -on**) and run an application that reserves any of the DM's color slots (slots 8-15, specified via ctm_$ calls), then attempting to disable monochrome mode (issuing a **mono -off** command) returns the following error:

"Current color map usage prevents turning MONO off."

OPTIONS

If you do not specify an option, **mono** toggles the current mode.

-on	Enable monochrome mode
-off	Disable monochrome mode

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help inv	For information on controlling window color on monochrome monitors
help bgc	For information on controlling background color on monochrome monitors

NAME

msg – display a message in the DM output window

SYNOPSIS

msg '*string*'

DESCRIPTION

The **msg** command instructs the DM to print a string in the DM output window. You must enclose the string in single quotation marks.

ARGUMENTS

string (required) Specify the string to be printed in the DM output window.

EXAMPLES

The DM command line

msg 'Please select another key'

causes the DM to display the message "Please select another key" in the DM output window.

NAME

pb – move bottom of pad into window

SYNOPSIS

pb

DESCRIPTION

The **pb** command moves the bottom line of the pad to the bottom of the current window. This is a pad movement command, as distinct from **tb**, which moves the cursor to the last line in the window, regardless of that line's position in the pad.

pb does not require either arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help pt	For details about moving the top line of a pad into a window
help dm commands	For a list of other pad control commands

NAME

ph – move pad horizontally by characters

SYNOPSIS

ph [*-n*]

DESCRIPTION

The **ph** command moves (scrolls) the pad horizontally under a window in units of characters.

By default, the boxed horizontal arrow keys scroll a pad in 10-character increments.

ARGUMENTS

[*-n*] Specify scrolling increment in characters. Positive (unsigned) *n* scrolls the pad left; negative *n* scrolls the pad right. Specifying **ph** with no argument scrolls the pad left one character; it is equivalent to specifying **ph 1**.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands	For a list of other pad control commands
help pp	For information on scrolling a pad vertically

NAME

pn – save transcript pad in named file

SYNOPSIS

pn *pathname*

DESCRIPTION

The **pn** command names a transcript pad and makes it permanent. That is, the pad is stored in a file and remains on the system after all windows to it are deleted. However, the file remains in use and locked, until the process is stopped and all windows are closed. If you do not use the **pn** command, transcript pads are deleted when all windows to them are deleted.

You can also use the **pn** command to change the name of an edit pad.

ARGUMENTS

pathname (required) Specify the pathname where the DM saves the pad. The pathname must be cataloged in a directory on your node (that is, you cannot save a file on your node if the filename is cataloged on some other node).

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other pad control commands

NAME

pp – scroll pad vertically by pages

SYNOPSIS

pp [*-n*]

DESCRIPTION

The **pp** command scrolls the pad vertically under a window in units of pages. By default, the boxed up and down arrow keys invoke this command, scrolling in half-page units. Specifying **pp** with no argument scrolls the file by one full page from the start to the end of the file.

ARGUMENTS

[*-n*] Specify scrolling increment in pages. Positive (unsigned) *n* scrolls down; negative *n* scrolls up. Note that *n* may also be a decimal fraction. A page is defined as the smaller of the following values:

- The number of lines that fit in the window
- The number of lines between the bottom of the window and the next form feed or frame

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help pv	For details about scrolling a pad vertically in units of lines
help dm commands	For a list of other pad control commands
help ph	For information on scrolling a pad horizontally

NAME

pt – move top of pad into window

SYNOPSIS

pt

DESCRIPTION

The **pt** command moves the top line of the pad to the top of the current window. This is a pad movement command, as distinct from **tt**, which moves the cursor to the first line in the window, regardless of that line's position in the pad.

pt does not require either arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help pb	For details about moving the bottom line of a pad into a window
help dm commands	For a list of other pad control commands

NAME

pv – scroll pad vertically by lines

SYNOPSIS

pv [*-n*]

DESCRIPTION

The **pv** command scrolls the pad vertically under a window in units of lines. By default, the shifted up and down arrow keys invoke this command, scrolling in one line units.

ARGUMENTS

[*-n*]	Specify the scrolling increment in lines. Positive (unsigned) *n* scrolls down; negative *n* scrolls up. Specifying **pv** with no argument scrolls the pad vertically by one line; it is equivalent to specifying **pv 1**.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help pp	For details about scrolling a pad vertically in units of pages
help dm commands	For a list of other pad control commands

NAME

pw – update edit file while maintaining edit pad unchanged

SYNOPSIS

pw

DESCRIPTION

The **pw** command updates a file that is being edited. It is valid only for writable edit pads. The first time you issue **pw**, the DM writes the contents of the edit pad to the file that is being edited, without closing the edit pad. The DM saves the previous contents of the file in a file with the same name and the suffix .bak. Subsequent **pw** or **wc** (window_close) commands rewrite the new file and leave the .bak version of the file unchanged.

pw differs from **wc** in that **pw** writes the file while **wc** closes the file, two distinct operations. **pw** leaves the edit pad open and it writes the new version of the file even if other windows are viewing the edit pad.

pw is useful if, for example, you want to try compiling a program you are editing. If you decide to make more changes to the program, you can just go back to the edit pad and continue editing, since updates made by **pw** leave the edit pad open and active.

The SAVE key executes **pw;ro** to save the pad and put it in read-only mode.

The **pw** command requires no arguments or options.

NAME

rm – replace a mark on the mark stack

SYNOPSIS

rm

DESCRIPTION

The **rm** command places the last issued mark (**dr**) back on the mark stack, allowing you to use the mark again.

rm requires no arguments or options.

NOTE

The mark stack is only two deep. Marks are removed when you execute an **xc** (cut) command and when you create a window. You can issue the **cms** (clear_mark_stack) command to clear the mark stack.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dr	For details about placing marks
help cms	For details about clearing the mark stack

NAME

ro – set read/write mode

SYNOPSIS

ro [-on | -off]

DESCRIPTION

The **ro** command puts a pad into (**-on**) and out of (**-off**) read-only mode. If you do not supply an option, the current mode is toggled. The pad must be in write mode (**-off**) in order for you to insert or delete anything.

An "R" appears in the window legend of a pad in read-only mode. The "R" disappears in write mode.

If you modify an edit pad, you must write it out with the **pw** command before you can make it read-only.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ei	For details about setting pad insert/overstrike mode
help ws	For details about setting window scroll mode
help wa	For details about setting window autohold mode
help wh	For details about setting window hold mode

NAME

rs – refresh screen

SYNOPSIS

rs

DESCRIPTION

The **rs** command refreshes the entire screen, updating all windows with any pending changes.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help rw	For details about refreshing windows
help dm commands	For a list of other display management commands

NAME

rw – refresh a window

SYNOPSIS

rw [-r]

DESCRIPTION

The **rw** command causes the DM to refresh the contents of the current window immediately, updating it with any pending changes.

When an unexpected system fault, such as a network failure, occurs, pads may be marked undisplayable in order to avoid further faults. When this happens, the DM displays an error message instead of the window's normal contents. When the problem has been resolved, use the **-r** (reset) option to redisplay the window's normal contents.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help rs	For details about refreshing the entire screen
help dm commands	For a list of other display management commands

NAME

s – substitute all occurrences of matched string in defined range

SYNOPSIS

[*range*] s [/*string1*/*string2*/]

DESCRIPTION

The s command substitutes one literal string for a string described by a regular expression over a defined text range. The command does not move the cursor or the window, but does update the window when the substitution is completed. Strings used with this command are also saved for later use (see below).

All substitutions are case sensitive, unlike searches, which ignore case unless told otherwise. You cannot disable substitution case sensitivity.

ARGUMENTS

If you do not specify an argument, the previous substitution is repeated from the current cursor position to the end of the line.

range (optional) Specify the range of text in which substitution is to be made.

Default if omitted: use the current cursor position to end of line

string1 (optional) Specify the string to be replaced in the form of a regular expression. If you omit this argument but use the opening delimiter (/) (for example, s//*string2*/), *string1* defaults to the string used in the last search operation. If you also omit the delimiter (for example, s/*string2*/), *string1* defaults to the string used in the last substitution operation.

Default if omitted: see above

string2 (optional) Specify a literal replacement string. (This is not a regular expression). You can use an "&" to denote *string1*. If *string1* is present, you must specify *string2*.

Default if omitted: repeat the last substitution command

EXAMPLES

Move to the first character in the pad, place a mark, and move to the last character in the pad.

<CMD> **s/Fielding/Tom Jones/**

Replace the string "Fielding" with "Tom Jones" throughout the marked range (in this case, the entire pad).

<CMD> **s/Tom/& Jones/**

Replace "Tom" with "Tom Jones". Because you did not mark or specify a range, the replacement takes effect from the current

cursor position to the end of the line.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm range	For details about text editing ranges
help dm commands	For a list of other pad editing commands
help patterns	For details about regular expressions

NAME

sc – set search case sensitivity

SYNOPSIS

sc [-on | -off]

DESCRIPTION

A search can be either case-sensitive or case-insensitive. In case-sensitive searches, the characters must match in case (that is, /mary/ does not locate the string "MARY"). In case-insensitive searches, uppercase and lowercase letters are considered equivalent. By default, searches are case-insensitive.

The **-on** option explicitly specifies a case-sensitive search; the **-off** option explicitly specifies a case-insensitive search. Typing the sc command without options toggles the current case comparison setting.

NOTE

The sc command has no effect on substitution operations, only on search operations. Substitutions are always sensitive to the case of the strings involved.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other pad editing commands

NAME

shut – shut down system

SYNOPSIS

shut [-f]

DESCRIPTION

The **shut** command exits from the DM and shuts down the system. The DM first closes all windows and pads, then unloads the operating system and enters the Mnemonic Debugger that resides in the node's boot programmable read-only memory (PROM). If user processes are still active, the **shut** command attempts to stop them. If they stop normally, the shutdown proceeds. If the DM cannot stop them normally, the **shut** command aborts.

Type **ex domain_os** in the Mnemonic Debugger to restart the system.

Specify the **-f** option to force either logout or shutdown. You can achieve the same effect by reponding "y" to a request to blast processes that cannot be closed normally. If you use this, however, remember that some disk space may be lost if processes cannot be terminated normally. Use the salvager **salvol** to recover the disk space,.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help ex	For details about exiting the DM to the boot shell without unloading Domain
help lo	For details about normal logout
help salvol	For details about recovering disk space

NAME

so – substitute first occurrence of matched string

SYNOPSIS

[*range*] so [*/string1/string2/*]

DESCRIPTION

The so command is identical to the s (substitute) command except that *string2* replaces only the first occurrence of *string1* in each line of the defined range of text.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help s	For details about the general substitute command
help patterns	For details about regular expressions
help dm range	For details about text editing ranges
help dm commands	For a list of other pad editing commands

NAME

sq – abort a search operation

SYNOPSIS

sq

DESCRIPTION

The **sq** command aborts a text search, and cancels any action involving the **echo** command. This command is equivalent to **abrt**.

sq aborts the current search. The DM returns the message "Search aborted." It does not move the window. Note that you cannot type this command during a search. You must invoke it with a defined key.

When you use **sq** to abort the **echo** command, **sq** cancels a move window with rubberbanding or grow window with rubberbanding operation; or it cancels highlighting for a defined range of text, depending on how you use **echo**.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other pad editing commands.

NAME

tb – move cursor to bottom line in window

SYNOPSIS

tb

DESCRIPTION

The **tb** command moves the cursor to the bottom line in the window. This is in contrast to the **pb** command, which moves the bottom line of the pad into the window.

tb requires no arguments or options.

NOTE

There is a homonymous shell command: **tb** (traceback) - print traceback after a fault.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tt	For details about moving the cursor to the top line in the window
help dm commands	For a list of other cursor movement commands

NAME

tdm – move cursor to DM input window

SYNOPSIS

tdm

DESCRIPTION

tdm moves the cursor to the DM input window (labeled "Command: " at the bottom of the screen) so that you can enter DM commands.

By default, the CMD key (L5) invokes the **tdm** command.

tdm requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other cursor movement commands

NAME

th – move cursor right to next tab stop

SYNOPSIS

th

DESCRIPTION

The **th** command moves the cursor right to the next horizontal tab stop. Tabs are global (that is, they apply to all windows); use the DM command **ts** to set them. Initially, tabs are set every five spaces.

By default, the TAB key invokes the **th** command. Note that this does not insert an ASCII tab character into the file; it simply positions the cursor at the next tab stop.

th requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help thl	For details about tabbing to the left
help ts	For details about setting tab stops
help dm commands	For a list of other cursor movement commands

NAME

thl – move cursor left to previous tab stop

SYNOPSIS

thl

DESCRIPTION

The **thl** command moves the cursor left to the next horizontal tab stop. Tabs are global (that is, they apply to all windows), and you can set them with the DM command **ts**. Initially, tabs are set every five spaces. **thl** requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help th	For details about tabbing to the right
help ts	For details about setting tab stops
help dm commands	For a list of other cursor movement commands

NAME

ti – move cursor to next input window

SYNOPSIS

ti

DESCRIPTION

ti moves the cursor to the next fully unobscured window in which input is accepted (that is, the next window that opens into neither a transcript nor a read-only edit pad). The cursor is placed at its last previous position in the window.

The DM scans the screen from left to right and top to bottom to find the next window.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help curs	For information on restricting cursor movement
help dm commands	For a list of other cursor positioning commands

NAME

tl – move cursor to the beginning of the current line

SYNOPSIS

tl

DESCRIPTION

The **tl** command moves the cursor left to the beginning of the current line. By default, the bar-left arrow key invokes the **tl** command.

tl requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tr	For details about moving to the (right) end of the current line
help dm commands	For a list of other cursor positioning commands

NAME

tlw – move cursor to last (previous) window

SYNOPSIS

tlw

DESCRIPTION

tlw moves the cursor back to the window it was in before it moved to the current window. The cursor is placed at its last previous position in the window.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other cursor positioning commands

NAME

tn – move cursor to next window

SYNOPSIS

tn

DESCRIPTION

tn moves the cursor to the next fully unobscured window on the screen. Any window that is partially covered by another is not considered in the search. The DM scans the screen from top to bottom to find the next window, selecting the one whose upper-left corner is the "highest" (that is, has the lowest y coordinate value), then proceeding downward across the screen. If there are panes within a window, the DM positions the cursor in the next lower pane until the pane choices are exhausted, before moving to the next "lower" window. Once the next window is located, the DM places the cursor at its last previous position within that window.

By default, the NEXT WNDW key (lb) invokes this command.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tni	For information about moving the cursor to the next fully unobscured icon on the screen
help dm commands	For a list of other cursor movement commands

NAME

tni – move cursor to next icon

SYNOPSIS

tni

DESCRIPTION

tni moves the cursor to the next fully unobscured icon on the screen. Any icon that is partially covered by another is not considered in the search. The DM scans the screen from top to bottom to find the next icon, selecting the one whose upper-left corner is the "highest" (that is, has the lowest y coordinate value), then proceeding downward across the screen.

This command is similar to the **tn** command, which positions the cursor at the next fully unobscured window on the screen.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tn	For information on moving the cursor to the next fully unobscured window on the screen
help tn	For a list of other cursor movement commands

NAME

tr – move cursor to the end of the current line

SYNOPSIS

tr

DESCRIPTION

The **tr** command moves the cursor right to the end of the current line. By default, the bar-right arrow key invokes the **tr** command.

tr requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tl	For details about moving to the beginning of the current line
help dm commands	For a list of other cursor positioning commands

NAME

ts – set tab stops for all windows

SYNOPSIS

ts [*n1*] [*n2*] ... [**-r**]

DESCRIPTION

The **ts** command sets the default tab stops for all windows. You can also set tab stops from within a program, using a call to the system routine pad_$set_tabs; tab stops set under program control override those set by **ts** within windows belonging to the program.

By default, tabs are initially set every five spaces.

The DM command = displays the line and column numbers of the current cursor position. This can be helpful when you are trying to set tab stops visually.

ARGUMENTS

If you do not specify an argument, **ts** sets a stop at every character on the line.

n1 n2 ... (optional) Specify tab stops. The *n* values are integers representing absolute character positions. You must specify them in increasing order. Column numbers start with one.

Default if omitted: see above

OPTIONS

-r Repeat the last interval.

NOTE

There is a homonymous shell command: **ts** – display the module name and time stamp. Type **help ts_dm** for details about that command.

EXAMPLES

ts 7 12 -r Set tabs at columns 7 and 12, and every five spaces thereafter.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help th For details about tabbing to the right

help thl For details about tabbing to the left

help dm commands For a topical index of DM commands

NAME

tt – move cursor to top line in window

SYNOPSIS

tt

DESCRIPTION

The **tt** command moves the cursor to the top line in the window. This is in contrast to the **pt** command, which moves the top line of the pad to the the top of the window.

tt requires no arguments or options.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help tb	For details about moving the cursor to the bottom line in the window
help dm commands	For a list of other cursor movement commands

NAME

twb – move cursor to a specified window border

SYNOPSIS

twb {-l | -r | -t | -b}

DESCRIPTION

The **twb** command moves the cursor to a border of the current window, as specified by the command options. You must specify an option with **twb**.

OPTIONS

You must specify one of the following options.

-l Move the cursor to the left window border parallel to the previous cursor position.

-r Move the cursor to the right window border parallel to the previous cursor position.

-t Move the cursor to the top window border directly above the previous cursor position.

-b Move the cursor to the bottom window border directly below the previous cursor position.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help dm commands For a list of other cursor movement commands

NAME

undo – undo previous DM command(s)

SYNOPSIS

undo

DESCRIPTION

undo works by compiling a history of DM activities in input and edit pads in reverse chronological order. Invoking **undo** reverses the effect of the most recent DM command. Use successive **undo**s to undo earlier commands. Note that this applies only to DM operations; you cannot undo shell operations, for example, compiling a program.

The **undo** buffers (one per edit pad and one per input pad) are circular lists that when full, eliminate the oldest entries to make room for new ones. Entries are grouped in sets. For example, an s (substitute) command may change five lines. While **undo** considers this to be five entries, the entries are grouped into a single set so that one **undo** changes all five lines back to their original state. When a buffer becomes full, the oldest set of entries is erased. This means that **undo** never partially undoes an operation: it either completely undoes the operation or does nothing.

An edit **undo** buffer can hold up to 1024 entries. An input **undo** buffer can hold up to 128 entries.

By default, the UNDO key invokes the **undo** command.

undo requires no arguments or options.

NAME

wa – set window autohold mode

SYNOPSIS

wa [-on | -off]

DESCRIPTION

The **wa** command switches a window into (**-on**) and out of (**-off**) autohold mode. **wa** without options toggles the current setting. In autohold mode, the window automatically enters hold mode (in which the contents of a window are temporarily frozen) if either of the following conditions is true:

- A full window of output is available and none of it has been displayed.
- A form feed or create frame operation is output to the pad. In this case, the window displays the output preceding the form feed. When the window exits from hold mode, the output following the form feed or create frame operation starts at the top of the window.

Initially, windows are not in autohold mode. The window legend contains an "A" when the window is in autohold mode.

NOTE

Autohold mode applies only to windows open into transcript pads.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wh	For details about window hold mode
help ws	For details about window scroll mode
help dm commands	For a list of other window management commands

NAME

wc – close window and associated functions

SYNOPSIS

wc [*entry_name*] [**-q** | **-f** | **-a** | **-s**]

DESCRIPTION

The **wc** command closes (deletes) a window or window group. It may also close the pad into which the window looks, depending on the following conditions.

If other windows into the pad besides the one being closed exist, the DM leaves the pad open. However, if there are no other windows into the pad, the DM closes it. The DM then either deletes the closed pad (if it was temporary) or saves it under its pathname (if it was named and permanent, that is, a permanent disk file).

If the pad is a writable edit pad, and is being viewed only through the current window, the DM renames the old file by appending .bak to its name, and writes the edited version to the original filename. If multiple windows are viewing the edit pad, **wc** simply closes the window, it does not write the file or rename the old file. To force the DM to write the file and create the .bak version, use the DM command **pw** (pad_write) or the EXIT, or SAVE keys (see below).

You normally cannot close a transcript (output) window if it is the last window into an active process (see **-f** below).

Note that the DM cannot delete a permanent pad (file).

Three keys are predefined to perform related functions:

pw;wc -q (or) <EXIT> (r5)	Close window, pad; update file
wc -q (or) <ABORT> (r5s)	Close window, pad; ignore changes
pw;ro (or) <SAVE> (r4s)	Save pad and put it in read-only mode; do not close it

ARGUMENTS

entry_name (optional)

Specify the name of the window or window group to be closed. If the name of the window or group appears as a text string somewhere on the display, you may use the following time-saving feature: place the cursor on the name, then press <MARK>. Now issue the **wc** command. **wc** uses the marked name for the *entry_name* argument.

Default if omitted: close the window under the cursor

OPTIONS

If you do not specify an option, wc closes the window and pad, then deletes the pad (if temporary) or rewrites it (if permanent) as described above. You can specify only one of the following options at a time.

-q Quit without updating the pad (file). The DM ignores any changes made while the window was open. The DM prompts you with "File modified. OK to quit?" if you made changes, to verify that you really wish to discard them.

-f Force window closure, even if this window is the last one open into a process. However, the process becomes inaccessible if no windows are left.

-a Enable auto-close for the current window. When auto-close is enabled, the current window closes when the pad into which it looks is closed.

-s Disable auto-close for the current window. If auto-close is disabled (the default condition), the current window persists after the pad into which it looks is closed.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows	For general information about windows
help dm commands	For a list of other window management commands

NAME

wdf – define DM default window positions

SYNOPSIS

[*region*] **wdf** [*n*]

DESCRIPTION

The **wdf** command lets you define any of the DM's twelve default window positions. To define a default window position, mark (with the DM command **dr**) the region that will display the window, and issue the **wdf** command.

ARGUMENTS

region (optional)	Specify the area of the screen where the new window will be displayed. Default if omitted: use marked region
n (optional)	Specify the ID number (1-12) of the DM default window that is being defined. If you omit *n*, **wdf** discards any saved window parameters, so that the next window created uses the stock default window boundaries. Default if omitted: see above

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows	For general information about windows
help idf	For information about defining DM default icon positions

NAME

wg – grow or shrink a window

SYNOPSIS

[*region*] **wg**

DESCRIPTION

The **wg** command changes the size of a window by moving one edge or corner across the screen while leaving the other edges and/or corners where they are. To grow or shrink a window, first mark the edge or corner you want to move by positioning the cursor at that edge or corner and issuing the **dr** command string or pressing <MARK>. Then move the cursor to the new location for the edge or corner and issue the **wg** command.

The marked edge or corner moves to the new cursor position, and the window shrinks or grows accordingly. If you want to move only an edge, move the cursor only in the direction perpendicular to that edge. Moving the cursor in two dimensions causes a corner to move.

ARGUMENTS

region (optional) Specify the old and new locations of edge or corner. You can do so in a variety of formats. See the help window for more information on format choices.

You must use this argument if you do not use the cursor placement and <MARK> operation described above.

Default if omitted: use marked region

NOTE

A companion grow command, **wge**, provides visible feedback during a grow operation.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows	For general information about windows
help dm commands	For a list of other window management commands

NAME

wge – grow/shrink a window with rubberbanding

SYNOPSIS

wge

DESCRIPTION

The **wge** command changes the size of a window. To enlarge or shrink a window with **wge**, position the cursor in the window and issue the **wge** command. The GROW key invokes the **wge** command by default. After you enter the **wge** command, an outline, or "rubberband" appears, to show you the size and shape that the window will take when you complete the grow operation. Move the cursor until the rubberband matches the new size you want for the window. Then issue the **dr echo** command sequence or press <MARK> to complete the grow operation. Use the **sq** command to abort a grow operation using rubberbanding.

wge requires no arguments or options.

NAME

wgra – create or add to a window group

SYNOPSIS

wgra *group_name* [*entry_name*]

DESCRIPTION

The **wgra** command creates a new window group with the specified *group_name*, or adds a window or group to an existing group. If you do not specify an *entry_name*, (the name of a window or group) **wgra** uses the name of the window where the cursor was last positioned.

ARGUMENTS

group_name (required)	Specify the name of the group to be created or enlarged.
entry_name (optional)	Specify the name of the window or group to be added to *group_name*. Default if omitted: use the name of the window where the cursor was last positioned

EXAMPLES

Command: **wgra Shell_Windows pad01**

This command adds a window called "pad01" to a group of windows called "Shell_Windows".

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wgrr	For details about removing windows from window groups
help windows	For general information about window groups

NAME

wgrr – remove window/group from group

SYNOPSIS

wgrr *group_name* [*entry_name*]

DESCRIPTION

The **wgrr** command removes a window or group from a window group. If you do not specify an *entry_name* (the name of a window or group) **wgrr** uses the name of the window where the cursor was last positioned.

ARGUMENTS

group_name (required)	Specify the name of the window group that contains the window or group you want to remove.
entry_name (optional)	Specify the name of the window or group to be removed. Default if omitted: use the name of the window where the cursor was last positioned

EXAMPLES

Command: **wgrr Shell_Windows pad02**

This command removes a window called "pad02" from a group of windows called "Shell_Windows".

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wgra	For details about creating and adding to window groups
help windows	For general information about window groups

NAME

wh – set window hold mode

SYNOPSIS

wh [-on | -off]

DESCRIPTION

The **wh** command switches a window into (**-on**) or out of (**-off**) hold mode. **wh** without options toggles the current setting. In hold mode, the contents of the window are frozen and do not change when a program sends more output to the pad. When a window is not in hold mode, the window automatically moves to the end of the pad as new output appears.

By default, the HOLD key invokes the **wh** command.

Initially, windows are not in hold mode. The window legend contains an "H" when the window is in hold mode.

NOTE

Hold mode applies only to windows open into transcript pads.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wa	For details about window autohold mode
help ws	For details about window scroll mode
help dm commands	For a list of other window management commands

NAME

wi – make a window or group visible or invisible

SYNOPSIS

wi [*entry_name*] [**-w** | **-i**]

DESCRIPTION

The **wi** command controls the visibility of the specified window or group. **wi** without options toggles the current mode. If you do not specify an *entry_name* (the name of a window or group) **wi** uses the name of the window under the cursor.

ARGUMENTS

entry_name (optional) Specify the name of the window or group you want to make visible or invisible. If the name of the window or group appears as a text string somewhere on the display, you can use the following time-saving feature: place the cursor on the name, then press <MARK>. Now issue the **wi** command. **wi** uses the marked name for the *entry_name* argument.

Default if omitted: manipulate the window under the cursor

OPTIONS

If you do not specify an option, **wi** toggles the current visiblity setting.

-i Force the window or group to be invisible.

-w Force the window or group to appear as a window.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows For details about window groups

help dm commands For a list of other window management commands

NAME

wm – move a window across the screen

SYNOPSIS

[*region*] **wm**

DESCRIPTION

The **wm** command moves a window across the screen. The DM moves the window whose nearest unobscured edge or corner is the first point of the region. The new location of this edge or corner is the second point of the region. Therefore, to move a window, place the cursor at one corner of the window you want to move and issue the **dr** command (or press <MARK>. Next, place the cursor at the desired new position of the corner. Finally, issue the **wm** command.

If you do not define a region, the **wm** command causes the nearest window corner to be moved to the current cursor position. However, this can cause unexpected results if there are multiple windows on the screen, since your idea of the nearest window may not be the same as the DM's. In that case, it is safer to mark the window you want to move.

ARGUMENTS

region (optional)	Specify the old and new locations of the edge or corner. You can do so in a variety of formats: see the help window for more information. Default if omitted: use the current cursor position

NOTE

A companion move command, **wme**, provides visible feedback during a move operation.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows	For general information about windows
help dm commands	For a list of other window management commands

NAME

wme – move a window using rubberbanding

SYNOPSIS

wme

DESCRIPTION

The **wme** command moves a window across the screen using the rubberbanding feature. To move a window, place the cursor in the window you want to move and issue the **wme** command. By default, the MOVE key invokes the **wme** command. After you issue the **wme** command an outline or "rubberband" appears, to show you where the window will be when you complete the move operation. Now move the cursor until the rubberband is at the desired window position. Finally, issue the **dr; echo** command or press <MARK> to complete the grow operation. You can use the **sq** command to abort a move operation using rubberbanding.

The **wme** command requires no arguments or options.

NAME

wp – push or pop a window on the stack

SYNOPSIS

wp [*entry_name*] [**-t**|**-b**]

DESCRIPTION

The **wp** command pops a window to the top of the stack or pushes a window to the bottom of the stack. If the cursor rests in a partially obscured window, the **wp** command pops the window to the top of the pile. If the cursor rests in a completely visible window, **wp** pushes the window to the bottom of the pile. **wp** can also manipulate specific named windows or window groups. See the arguments section below.

By default the POP key invokes the **wp** command.

ARGUMENTS

entry_name, (optional) Specify the name of the window or group you want to push or pop. If the name of the window or group appears as a text string somewhere on the display, you may use the following time-saving feature: place the cursor on the name, then press <MARK>. Now issue the **wp** command. **wp** uses the marked name for the *entry_name* argument.

Default if omitted: push or pop the window under the cursor

OPTIONS

The following options are intended primarily for use in DM scripts, where you may not be able to predict the presence of other windows on the screen.

-t Force a window to the top of the window stack.

-b Force a window to the bottom of the window stack.

EXAMPLES

wp -t Pop the window containing the cursor to the top of the window stack.

wp slide -b Push the window named 'slide' to the bottom of the stack.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help windows For general information about windows

help dm commands For a list of other window management commands

NAME

ws – set window scroll mode

SYNOPSIS

ws [-on | -off]

DESCRIPTION

The **ws** command switches a window into (**-on**) and out of (**-off**) line-at-a-time scrolling. **ws** without options toggles the current setting. When line-at-a-time scrolling is in effect, output appears in the window one line at a time, scrolling past. When line-at-a-time scrolling is not in effect, output appears a window at a time.

Initially, all windows (except edit windows) have line-at-a-time scrolling. The window legend contains an "S" when the window is in scroll mode.

NOTE

Scroll mode applies only to windows open into transcript pads.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help wh	For details about window hold mode
help wa	For details about window autohold mode
help dm commands	For a list of other window management commands

NAME

xc – copy text to paste buffer

SYNOPSIS

[*range*] **xc** [**-r**] [**-f** *pathname* / *name*]

DESCRIPTION

The **xc** command copies a range of text from any pad into a paste buffer or system file. The copied text remains undisturbed.

By default, the COPY key invokes the **xc** command, using the default (unnamed) paste buffer.

ARGUMENTS

range (required) Specify the range of text to be copied.

Default if omitted: copy from cursor to end of line

name (optional) Specify the paste buffer name. Text is written to the named buffer. If text is copied to a buffer that was previously used, the new text overwrites the old. You can have up to 100 buffers open per log-in session.

Default if omitted: use **'node_data/paste_buffers/default.txt**

OPTIONS

-f *pathname* Specify the system file to receive copied text. If the file already exists, the copied text overwrites the current file contents. This option is not valid if the *name* argument is present.

-r Specify copy for a rectangular portion of text.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help xd For details about cutting text to a paste buffer

help xp For details about pasting in buffered text

help dm commands For a list of other pad editing commands

NAME

xd – cut (delete) text and write it to paste buffer

SYNOPSIS

[*range*] **xd** [**-r**] [**-f** *pathname* | *name*]

DESCRIPTION

The **xd** command copies a range of text into a paste buffer or system file, then deletes the text from the pad. You can use this command only in a writable pad.

By default, the CUT key invokes the **xd** command using the default (unnamed) paste buffer.

ARGUMENTS

range (required) Specify the range of text to be cut.

Default if omitted: cut from the cursor to the end of the line

name (optional) Specify the paste buffer name. **xd** writes the text to the named buffer. If you specify a buffer that was previously used, the new text overwrites the old. You can have up to 100 buffers open per log-in session.

Default if omitted: use **'node_data/default.txt**

OPTIONS

-f *pathname* Specify the system file to receive cut text. If the file already exists, the cut text overwrites the current file contents. This option is not valid if a *name* argument is present.

-r Specify a cut for a rectangular portion of text.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help xc For details about copying text to a paste buffer

help xp For details about pasting in buffered text

help dm commands For a list of other pad editing commands

NAME

xi – copy a display image into a graphics map file

SYNOPSIS

[*range*] **xi** [**-f** *pathname*]

DESCRIPTION

The **xi** command copies a display image into a graphics map file (GMF). If you do not mark the portion of the display window you want to copy, **xi** copies the entire window where the cursor is positioned. Use the **prf** shell command with the **-plot** option to print the GMF.

ARGUMENTS

range (optional)	Specify the range of image to be copied.
	Default if omitted: copy current window

OPTIONS

-f *pathname*	Specify the GMF output file. If you omit this option, **xi** writes the image to **'node_data/paste_buffers/default.gmf**.

NOTE

On a color node, only the text plane (plane 0) is copied. A color image is not copied.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help xc	For details about copying text to a paste buffer
help cpscr	For details about copying the entire screen to a GMF file
help prf	For details about printing files

NAME

xp – paste (write) buffered text into pad

SYNOPSIS

xp [-r] [-f *pathname* | *name*]

DESCRIPTION

The **xp** command inserts the contents of a paste buffer or system file into a pad at the current cursor position. **xp** does not change the contents of the paste buffer or file, so you can make multiple insertions. You can use this command only in a writable pad.

By default, the PASTE key invokes the **xp** command, using the default (unnamed) paste buffer.

ARGUMENTS

name (optional)	Specify the paste buffer name. **xp** copies text from the named buffer. You can have up to 100 buffers open per log-in session. Default if omitted: use **'node_data/paste_buffers/default.txt**

OPTIONS

-f *pathname*	Specify the system file to provide paste text. This option is not valid if *name* argument is present.
-r	Specify paste of a rectangular portion of text.

SEE ALSO

More information is available. Type the following at an Aegis shell prompt:

help xc	For details about copying text to a paste buffer without deleting it from the pad
help xd	For details about cutting text to a paste buffer
help dm commands	For a list of other pad editing commands

Reader's Response

Please take a few minutes to send us the information we need to revise and improve our manuals from your point of view.

Document Title: *Domain/OS Display Manager Command Reference*
Order No.: 011418-A00
Date of Publication: July, 1988

What type of user are you?

_____ System programmer; language _____
_____ Applications programmer; language _____
_____ System maintenance person
_____ System Administrator
_____ Student Programmer
_____ Other

_____ Manager/Professional
_____ Technical Professional
_____ Novice

How often do you use the Apollo system?_____

What parts of the manual are especially useful for the job you are doing?_____

What additional information would you like the manual to include?_____

Please list any errors, omissions, or problem areas in the manual. (Identify errors by page, section, figure, or table number wherever possible. Specify additional index entries.)_____

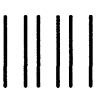
Your Name Date

Organization

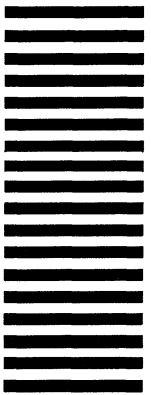
Street Address

City State Zip

No postage necessary if mailed in the U.S.



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824

Reader's Response

Please take a few minutes to send us the information we need to revise and improve our manuals from your point of view.

Document Title: *Domain/OS Display Manager Command Reference*
Order No.: 011418-A00
Date of Publication: July, 1988

What type of user are you?

_____ System programmer; language _____
_____ Applications programmer; language _____
_____ System maintenance person
_____ System Administrator
_____ Student Programmer
_____ Other

_____ Manager/Professional
_____ Technical Professional
_____ Novice

How often do you use the Apollo system?_____

What parts of the manual are especially useful for the job you are doing?_____

What additional information would you like the manual to include?_____

Please list any errors, omissions, or problem areas in the manual. (Identify errors by page, section, figure, or table number wherever possible. Specify additional index entries.)_____

Your Name Date

Organization

Street Address

City State Zip

No postage necessary if mailed in the U.S.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824

Reader's Response

Please take a few minutes to send us the information we need to revise and improve our manuals from your point of view.

Document Title: *Domain/OS Display Manager Command Reference*
Order No.: 011418-A00
Date of Publication: July, 1988

What type of user are you?

_____ System programmer; language _____
_____ Applications programmer; language _____
_____ System maintenance person
_____ System Administrator
_____ Student Programmer
_____ Other

_____ Manager/Professional
_____ Technical Professional
_____ Novice

How often do you use the Apollo system?_____

What parts of the manual are especially useful for the job you are doing?_____

What additional information would you like the manual to include?_____

Please list any errors, omissions, or problem areas in the manual. (Identify errors by page, section, figure, or table number wherever possible. Specify additional index entries.)_____

Your Name Date

Organization

Street Address

City State Zip

No postage necessary if mailed in the U.S.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824

apollo

Domain/OS Display Manager Command Reference

011418-A00