

Friday December 28, 2012

dL4Term Terminal Emulator 10.1.4 Maintenance Release
All Rights Reserved. Copyright (c) 1999 - 2012 by:
Dynamic Concepts, Inc. Aliso Viejo, CA 92656 USA

Email address: techsupport@dynamic.com
Information: www.dynamic.com
Downloads: ftp.dynamic.com

Highlights of This Release =====

- o GUI grid controls.
- o GUI date controls.
- o WCACTION code 6 to fully distinguish double clicking from single clicking.

What is dL4Term? =====

- o dL4Term is an SSH/telnet client which is installed on Windows 98/ME or Windows NT/2000/XP/2003/2008/Vista systems. Unlike other SSH/telnet programs, dL4Term does not emulate a specific character oriented terminal such as a Wyse60 or VT100. Instead, it offers its own set of character based and GUI (graphical user interface) features. The feature set is the set of UniBasic/dL4 mnemonics, including colors, fonts, protection and dL4 version 4 GUI mnemonics.

dL4Term operates on the same principle as other emulators in that it recognizes specific character sequences to invoke its features. It differs in the sense that it cannot be instructed to recognize, and thereby emulate, the hard-coded character sequences of specific terminals. When connected to a Unix host, for example, dL4Term supports shell, vi, IQ, UniBasic and dL4. In essence, dL4Term is a GUI (graphic user interface) oriented terminal model with multiple windows and graphic controls such as buttons and list boxes. It also supports displaying multiple fonts or graphics in any of the windows belonging to the GUI "terminal".

Using dL4Term, GUI programs can run under either dL4 for Windows or dL4 for Unix and still produce the same Windows style user interface. These GUI features can also be used without using dL4. All of the GUI functions are controlled by character sequences thus allowing any Unix application to produce and control a Windows-like user interface. For character oriented Unix applications such as "vi", the GUI terminal model includes all necessary character terminal functions such as cursor positioning, line insertion/deletion, and protected characters.

dL4Term can be installed and used on any Windows NT/2000/XP/2003/2008/Vista/7 system to support SSH or telnet (TCP/IP) connections to a Unix host. The standard SSH or telnet protocol, as selected by the user, is used to communicate with host systems. Unix terminal definition files are supplied and must be installed on each Unix host before using vi (terminfo), dL4 for Unix, UniBasic, or IQ for Unix.

For GUI programming in UniBasic and languages other than dL4, the dL4Term user guide describes each of the character sequences used by dL4Term. GUI mnemonic use in general is described in the "dL4 4.3 GUI Training" document. Both documents are available on www.dynamic.com and ftp.dynamic.com.

Acknowledgements

=====

The SSH protocol support in dL4Term is derived from the OpenSSH package in accordance with the OpenSSH license. The SSH support functions are linked with the OpenSSL library. The OpenSSH and OpenSSL license files and copyright notices are included in the file credits.txt.

Installation instructions =====

To install dL4Term for Windows, perform the following steps:

1. Exit from and terminate any running instances of dL4Term for Windows.
2. Remove (uninstall) any previous dL4Term for Windows release.
3. Run the dL4Term installation program 1B_dl4term_10.1.4.exe and follow the displayed instructions.
4. Install terminal definitions for dL4Term on the host Unix system as described in the next section.

Before using dL4Term, terminal definitions must be installed on the host Unix system. On each Unix host system, the system administrator should complete the following steps:

1. Copy each of the files from the dL4Term Samples directory to the Unix host system using FTP BINARY mode or any other method that will copy the files verbatim without a conversion from DOS to Unix text format.
2. As root, compile the file dl4term.tic using the following command:

```
tic dl4term.tic
```

This command will compile and install a terminal definition for use by "vi" and other "curses" based programs that use terminfo. On Linux systems that use "termcap" for some utilities, the command

```
tic -C dl4term.tic >>/etc/termcap
```

should be used to translate dl4term.tic to termcap format and append it to the termcap database. The file dl4term.cap can also be used with termcap implementations, but it may need modifications to avoid syntax errors. Some terminal dependent applications such as "bash" may require application dependent configuration. The "bash" shell, for example, often requires defining function keys in a ".inputrc" home directory file for any terminal which is not using VT100 function keys. The following ".inputrc" file enables arrow, home, and end keys needed for basic line editing or using the command history:

```
$include /etc/inputrc
"\C-aF088": backward-char
"\C-aF0A0": forward-char
"\C-aF08F": beginning-of-line
"\C-aF10A": end-of-line
"\C-aF0AA": next-history
"\C-aF0AB": previous-history
```

The key definitions can also be added directly to /etc/inputrc.

3. To use dL4 for Unix with dL4Term, copy the file dl4term to the directory /usr/lib/dl4/term. After copying the file, use the following command to make it readable by all users:

```
chmod 444 /usr/lib/dl4/term/dl4term
```

If the dL4 terminal definition files were installed in a directory other than /usr/lib/dl4/term, simply change the commands to use the actual directory. Note that this terminal definition file can only be used with version 4.1 or later of dL4 for Unix.

To use dL4Term with dL4 3.x or earlier revisions, copy the file dl4term.dl43 to /usr/lib/dl4/term/dl4term.

4. To use UniBasic with dL4Term, copy the file term.dl4term.iris or term.dl4term.bits to /usr/lib/ub/sys/term.dl4term. If applications use the 'FM' and 'FX' mnemonics to enable and disable protected characters, copy the file term.dl4term.iris. If applications expect the 'BP' mnemonic to both paint and enable protected characters, copy the file term.dl4term.bits. After copying the file, use the following command to make it readable by all users:

```
chmod 444 /usr/lib/ub/sys/term.dl4term
```

If the UniBasic terminal translation files were installed in a directory other than /usr/lib/ub/sys, simply change the commands to use the actual directory.

5. To use IQ for Unix with dL4Term, execute the following command to append the dL4Term definition to iqcap:

```
cat dl4term.iqcap >>/usr/lib/iq4/iqcap
```

If IQ for Unix was installed in a directory other than /usr/lib/iq4, simply change the command above to use the actual directory.

6. On most Unix host systems, the environment variable TERM will be automatically set to the required value of "dl4term" when a dL4Term user logs onto the system. If not, the user ".profile" or other shell initialization file should be modified to set TERM to "dl4term".

File list =====

readme.txt	This file
license.txt	Runtime license terms and conditions
credits.txt	OpenSSH and OpenSSL license and copyright information
dl4ssh.dll	Runtime library
dl4rt.dll	Runtime library
dl4stdrv.dll	Runtime library
dl4sockt.dll	Runtime library
msvcrt.dll	Runtime library
dl4term.exe	dL4Term executable
stubwa10.dll	Runtime library
Samples\	Directory of terminal definition files for host systems
	All of these files are Unix format text files and may not be displayable under Windows.
dl4term.tic	TermInfo source file for dL4Term terminal definition
dl4term	dL4 Terminal Description File (for dl4 5.1 or later)
	This file is recommended for use on most systems.
dl4term.dl43	dL4 Terminal Description File (for dl4 3.x or earlier)
	To use this file, it should be renamed as "dl4term".
dl4term.chwin	dL4 Terminal Description File for dl4 4.1 or later
	using the character window driver instead of the standard GUI window driver. To use this file, it should be renamed as "dl4term". This file might be needed if applications depend on character terminal oriented behavior of the window driver.
dl4term.iris	UniBasic terminal translation file with IRIS style treatment of protected characters
dl4term.bits	UniBasic terminal translation file with BITS style

dl4term.iqcap	treatment of protected characters
dl4term.cap	IQ for Unix IQCAP entry for dL4Term
	Termcap source file for dL4Term terminal definition.
	This file was prepared for Linux termcap, but may
	be usable with modification in other termcap
	implementations.

The following font files are installed for use with dL4Term:

arialmon.ttf	Monospaced Arial font
arialmb.ttf	Monospaced Bold Arial font

New in This Release

=====

Dec 28, 2012 (Maintenance Release 10.1.4)

- o Bug fixed: tabbing between GUI elements did not work in a child window that contained a WCPAD element with tip text.
- o Bug fixed: WCGRID and WCDATE GUI elements were not displayed as disabled if they were created with the "disabled" option (option 1).

Nov 21, 2012 (Maintenance Release 10.1.3)

- o Bug fixed: in 'WCGRID' or 'WCSORTGRID' GUI elements, some column types, such as image buttons, were not displayed as disabled when the grid element itself was disabled.
- o Bug fixed: 'WCDATE' GUI elements did not properly display themselves as disabled when the elements were disabled.

Nov 16, 2012 (Maintenance Release 10.1.2)

- o Enhancement: the "mask\$" parameter in a 'WCGRID' column definition can now be used to set the maximum number of characters in edit box columns (such as 'WCSTRING') by using a numeric value instead of a string.

The WCGRID "columns" string is a sequence of column definition strings. Each column definition string has the format:

```
PChr$(width, name$)+'WCxxxx'
PChr$(width, name$, options)+'WCxxxx'
PChr$(width, name$, options, mask$)+'WCxxxx'
PChr$(width, name$, options, maxlen)+'WCxxxx'
```

width	initial column width in grid column units
name\$	column label to be displayed in the header.
options	an optional numeric parameter, see description below.
mask\$	optional string parameter controlling the display format (not supported in this release).
maxlen	optional numeric parameter controlling the maximum length in characters of a 'WCSTRING', 'WCNUMBER', 'WCPRIVATE', or 'WCEDITDROP' column value.
WCxxxx	column type specified as 'WCSTRING', 'WCNUMBER', 'WCDATE', 'WCPRIVATE', 'WCTEXT', 'WCLISTDROP', 'WCEDITDROP', 'WCBUTTON', 'WCRJBUTTON', 'WCLJBUTTON', 'WCCHECK', or 'WCRADIO'.

- o Behavior change: 'WCDATE' GUI elements and grid columns now allow users to enter a date as a numeric string without separators. For example, if the system date format is MM/DD/YY, a user could enter "12382" or "01231982" for January 23, 1982.
- o Behavior change: 'WCTEXT' columns in 'WCGRID' GUI elements now return a value of "" instead of the displayed value of the read-only column. This is consistent with the behavior of 'WCTEXT' GUI elements and reduces network traffic.

- o Bug fixed: if option 4 (send "change" event) was used with a 'WCDATE' GUI element, the change event string was not sent if the date was changed using the dropdown calendar.
- o Bug fixed: 'WCBQRYBUF' set the special query buffering mode only in the window where the 'WCBQRYBUF' mnemonic was printed instead of setting the mode for all windows as documented.
- o Bug fixed: 'WCTEXT' columns in 'WCGRID' GUI elements did not display a value.

Oct 12, 2012 (Maintenance Release 10.1.1)

- o The SSH user/host syntax has been enhanced to interpret "@hostname" as "user@hostname" where the user name is taken from the "USERNAME" environment variable.
- o The dL4Term command line options have been extended with "-e" to select "Exit on Disconnect" behavior and "-q" to select "Disconnect Silently" behavior.
- o The mnemonic sequence 'OENQ' has been implemented to return a client operating system identification string followed by an optional string from the registry and a carriage return. The operating system string has the format "MSWmm.nn;" where "MSW" indicates Microsoft Windows, "mm" is the major version number, and "nn" is the minor version number. The version numbers are defined by Microsoft as shown below:

MSW6.2	Windows 8 or Windows Server 2012
MSW6.1	Windows 7 or Windows Server 2008R2
MSW6.0	Windows Vista or Windows Server 2008
MSW5.2	Windows XP 64-bit or Windows 2003
MSW5.1	Windows XP
MSW5.0	Windows 2000

If a registry string value is defined in

HKEY_LOCAL_MACHINE\Software\DynamicConcepts\dl4Term\WinTerm\ENQString

or

HKEY_CURRENT_USER\Software\DynamicConcepts\dl4Term\WinTerm\ENQString

that string will be appended to the "MSWmm.nn;" string. Any non-printable ASCII characters in the string will be replaced with spaces.

Sep 5, 2012 (Release 10.1)

- o New GUI mnemonics, 'WCGRID' and 'WCSORTGRID', have been implemented to create grid style GUI input elements. The 'WCGRID' mnemonic is used to create a grid of unlimited size where the application implements any sorting options. The 'WCSORTGRID' mnemonic is used for smaller grids where the user can change the displayed order at any time by "clicking" on a column header. Grid boxes consist of a header row and one or more data rows. The rows can contain one or two lines. The header row displays labels for each column and shows which column, if any, is used for the sorting order (ascending or descending). The header column separators can be "grabbed" using a left mouse click and used to resize columns. Individual columns can be string boxes, number boxes, date boxes, private string boxes, list drop boxes, edit drop boxes, buttons, check boxes, or radio button. The 'WCGRID' and 'WCSORTGRID' mnemonics are supported in dL4 for Unix/Linux 10.2 and in dL4 for Windows 10.1

Vertical and horizontal scroll bars are provided if the number of rows or the row width exceeds the size of the grid box. Users can also scroll through the grid by typing up arrow, down arrow, page up, or page down keys in a grid cell.

Syntax:

```
PChr$(n,x1, y1, x2, y2, "columns", options, rowh);'WCGRID'  
PChr$(n,x1, y1, x2, y2, "columns", options, rowh, headerh);'WCGRID'
```

n element number of the grid box (must be unique in window)
x1,y1 Upper left grid coordinates of the rectangle containing the
 grid box and title
x2,y2 Lower right grid coordinates of the rectangle enclosing the grid
 box and title
columns a string that defines the columns and column labels.
options a required numeric parameter, see description below
rowh numeric data row height in grid coordinates.
headerh an optional numeric parameter specifying the height of the header
 row. If not specified, the data row height is used.

The "columns" string is a sequence of column definition strings. Each column definition string has the format:

```
PChr$(width, name$)+'WCxxxx'  
PChr$(width, name$, options)+'WCxxxx'  
PChr$(width, name$, options, mask$)+'WCxxxx'
```

width initial column width in grid column units
name\$ column label to be displayed in the header.
options an optional numeric parameter, see description below.
mask\$ optional string parameter controlling the display format (not
 supported in this release).
WCxxxx column type specified as 'WCSTRING', 'WCNUMBER', 'WCDATE',
 'WCPRIVATE', 'WCTEXT', 'WCLISTDROP', 'WCEDITDROP', 'WCBUTTON',
 'WCRJBUTTON', 'WCLJBUTTON', 'WCCHECK', or 'WCRADIO'.

If a long row is to be displayed as two lines, a 'CR' mnemonic must be used in the 'columns' string as a line delimiter.

Radio buttons must be adjacent to form a group. Radio button groups must be separated by a non-radio button column.

The grid "options" parameter is the sum of the following option values:

1 Disable and "gray out" the grid box (user can't select the box)
2 Use the grid box as a tab stop (in element number order)
4 Send input to window when the user leaves a modified row
 MSC(48) will be set to the row number and MSC(47) will be -1
16 Don't display first field of row, but return it as part of the row
32 Any loss of input focus is reported as an input value change

If not needed, the value of the grid "options" must be zero.

The column "options" parameter is the sum of the following option values:

4 Send input to window when the user leaves a modified cell
 MSC(48) will be set to the row number and MSC(47) will be the
 column number.
16 Don't display first field of drop down list, but return it as the
 value of the cell (WCLISTDROP columns only)
32 Any loss of input focus is reported as an input value change

If not needed, the value of the column "options" must be zero.

As in 'WCLISTGRID' boxes, WCACTIONS 4 and 5 are supported to tell the application when a user has scrolled to the beginning or end of the grid so that the application can add new grid rows.

Rows are inserted into a grid by printing the column values as strings with 'HT' mnemonics terminating each column. The final column of a row can be terminated by a 'CR' mnemonic or end-of-line character. The row

is inserted at the current position which is typically the end of the grid but may be changed by printing 'n,r WCSELECT' where 'n' is the GUI element number and 'r' is the grid row number (0 origin). If the 'MH' mnemonic is the first character of a new row, that row will be inserted at the beginning of the grid instead of the end of the grid without changing the current position. An existing row can be deleting by selecting the row using 'n,r WCSELECT' and then printing a line containing only a 'DL' mnemonic. An existing row can be updated by selecting the row and then printing the new row contents as a line prefixed with a 'DL' mnemonic.

The 'n WCMARK' and 'n WCUNMARK' mnemonics can be printed to a grid element to set a row as currently selected or to clear the selection. The 'fg,bg WCMARKCOLOR' mnemonic can be printed to a grid element to set the color used to display the currently selected row.

The 'n,r,c WCFOCUS' mnemonic can be printed to set the input focus on row 'r' and column 'c' of the grid element 'n'.

The 'PREVPAGE' and 'NEXTPAGE' mnemonics can be used to define popup messages to inform the user that the end of the grid has been reached or to wait while the application is reading additional rows. The message text is set by printing a line to the grid with a 'PREVPAGE' or 'NEXTPAGE' mnemonic as the first character of the text. Any existing 'PREVPAGE' or 'NEXTPAGE' message will be replaced. These messages can be deleted by printing a line that contains only a 'PREVPAGE' or 'NEXTPAGE' mnemonic.

Immediately after a grid element is created, the drop-down lists in 'WCLISTDROP' or 'WCEDITDROP' columns must be initialized by printing each list ending with a double 'CR'. If a 'WCLISTDROP' column used option 16 ("hidden" field), then each line must consist of the hidden value, an 'HT' character, and the associated display value.

To display grid rows using alternating colors, print a 'fg,bg WCALTCOLOR' mnemonic to the grid.

To display modified cells with specific colors, print a 'fg,bg WCEDITCOLOR' mnemonic to the grid.

To set tool tip text for each cell in a column, print

```
PChr$(n, c, "text");'WCTIPTEXT';
```

where 'n' is the element number of the grid and 'c' is the column number starting at 0.

When using 'WCGRID', the column headers act as buttons. If "clicked", the application receives a function key character ('Fn' where 'n' is the grid element number modulo 64) or, if set, the action string. The application can determine which header was selected by using the MSC(47) function to determine the zero based column number (MSC(48), the row number, will be -1). The application can then ignore the request or reload the grid in the requested order. An 'n,c,d SORT' mnemonic can be printed to display an up arrow (ascending) or down arrow (descending) indicator on header column 'c' of the grid 'n' ('d' is 1 for ascending order or -1 for descending order).

When using 'WCSORTGRID', the column headers act as locally processed sort control buttons. The display order is changed immediately after the user "clicks" on the header. The application is not informed about the change and will continue to use and see the original row numbers used to populate the grid. If rows are added by the application to the grid after the user selects a sorting order, the new rows will be displayed in the selected order. If the user modifies a row after selecting a sorting order, the row will NOT be repositioned unless the sorting order is changed. An application can select a sorting order by printing the 'n,c,d SORT' mnemonic where 'n' is the element number of the grid, 'c' is the zero based column for sorting, and 'd' is 1 for ascending order or -1 for

descending order.

If a button in a row is "clicked", the application receives a function key character ('Fn' where 'n' is the grid element number modulo 64) or, if set, the action string. The application can determine which button was selected by using the MSC(47) function to determine the zero based column number and MSC(48) to get the row number.

Users can enter data in any column other than a WCTEXT column or a push button if the column already has a value. To allow a user to define a new row, the row must be filled with default values.

Grid elements do not return anything if a 'WCQUERY' mnemonic is sent to a grid element. The values of individual rows can be requested by printing the 'n,r WCQUERYROW' mnemonic where 'n' is the grid element number and 'r' is the desired row. The value of the row is returned as a carriage return terminated line with tab characters ('HT') separating the column values. Applications can parse the result string themselves or use the new intrinsic CALL GRIDROW() to store the column values. An application can request the values of modified or new rows by printing the 'n,0 WCQUERYNEW' mnemonic which returns the first such row or an empty line if there are no such rows. Additional modified or new rows can be requested by printing the 'n WCQUERYNEW' mnemonic. All rows returned by 'WCQUERYNEW' are prefixed by the row number and a tab ('HT') character.

As with all other GUI mnemonics, the 'WCGRID' and 'WCSORTGRID' mnemonics can only be used with dL4Term or in dL4 for Windows. The grid mnemonics cannot be used on Windows 98 or ME.

Example program:

```
Declare Intrinsic Sub GridRow
Def Struct ROW=Name$(30),3%,Cost,Code$(20)
Dim Header$(100),Action$(1),Query$(2000),Row. As ROW
Print 'XX 0 FONTCOLOR 12632256 BACKCOLOR CS';
Print PChr$(0,2^24-1);'WCSETCOLOR';
Print 'WCSETFONT';
! Create grid element
Header$ = PChr$(20,"Product name") + 'WCSTRING'
Header$ = Header$ + PChr$(10,"Cost") + 'WCNUMBER'
Header$ = Header$ + PChr$(10,"Code") + 'WCEDITDROP'
Print PChr$(2,10,5,72,10,Header$,4,1);'WCSORTGRID';
Print '2WCSELECT';
Print PChr$(0,255*256);'WCALTCOLOR';
! Initialize the "Code" dropdown list
For I = 1 To 5
    Print I Using "Code&&"
Next I
Print
! Place initial data in grid
Print 'PREVPAGE' + "Please wait, reading preceding page."
P = 1000
N = 1019
For I = P To N
    Print "Name #";I;'HT';(I + .99) Using "#####.##";'HT';BStr$(I * 12345,16)
Next I
Print 'NEXTPAGE' + "Please wait, reading next page."
! Set initial position to middle of data
Print '10WCMARK';
Print '0WCSELECT';
! Set scrolling actions
Print PChr$(2,4,'F20');'WCACTION';
Print PChr$(2,5,'F21');'WCACTION';
Print PChr$(1,36,12,44,14,"Done",0);'WCBUTTON'; ! display 'Done' button
Print @10,1;"Modify one or more rows and then press the 'Done' button."
Print 'BQ';
Do
    Input Len (1);""Action$
```



```

If Action$ = 'F1'
    Print @0,17;"Done button pressed!"
    Exit Do
End If
Select Case Action$
    Case 'F2'
        Print @0,3;"Row";Msc(48);"changed"
    Case 'F20'
        ! User is at first row, add more at the start of the grid
        Print '2WCSELECT';
        For I = 1 To 10
            P = P - 1
            If P <= 900
                ! Simulate beginning of file and delete "reading" message
                Print 'PREVPAGE';"Beginning of table"
                Exit For
            End If
            Print 'MH';
            Print "Name #";P;'HT';(P + .99) Using "#####.##";'HT';
            Print BStr$(P * 12345,16)
        Next I
        Print '0WCSELECT';
    Case 'F21'
        ! User is at last grid row, add more at the end of the grid
        Print '2WCSELECT';
        For I = 1 To 10
            N = N + 1
            If N >= 1100
                ! Simulate end of file and replace "reading" message
                Print 'NEXTPAGE';"End of table"
                Exit For
            End If
            Print "Name #";N;'HT';(N + .99) Using "#####.##";'HT';
            Print BStr$(N * 12345,16)
        Next I
        Print '0WCSELECT';
    Case Else
        Print "Error input";BStr$(Int(Action$),16)
End Select
Loop
Print @0,18;"New or modified rows"
Print '2,0WCQUERYNEW';
Do
    Input ""Query$
    If Query$ = "" Exit Do
    Call GridRow(Query$,RowNum,Row.)
    Print "Changed row";RowNum;"    Name = ";Row.Name$;"    Cost =";Row.Cost;
    Print "    Code = ";Row.Code$
    Print '2WCQUERYNEW';
Loop
Print 'EQ 0WCFOCUS K3';
End

```

- o A new GUI mnemonic, 'WCDATE', has been implemented to create GUI input elements for date values. A date input box consists of an edit box similar to 'WCNUMBER' and a drop down button. The edit box is used to display and input date values. The drop down button, when pressed, puts a calendar under the edit box which can be used to select a new date. The edit box displays the date value in a numeric format such as "MM/DD/YYYY" with the ordering of year, month, and day as well as the separator character controlled by the Windows language/region settings. User input is restricted to digits, separator characters, and edit keys. The date value can be set by writing a date in the format "YYYYMMDD" followed by a carriage return to the date element. The date can be set to nothing (the initial default value) by writing "" to the date element or by the user deleting all characters from the edit box. When queried, a date element returns the date value in the format "YYYYMMDD" if the date is set, "" if no date is set, and "00000000" if the date is set but

invalid.

- o A new 'WCACTION' code, 6, has been implemented to report double clicking on list box elements or buttons without reporting a preceding single click code. This makes it possible for an application to use double clicks for a completely different function than single clicks. WCACTION 6 is supported for use with 'WCBUTTON', 'WCLIST', 'WCLISTGRID', 'WCSHOWLIST', 'WCSHOWLISTGRID', and 'WCPAD'.

Example:

```
PRINT PCHR$(10,6,'F40');'WCACTION';
```

Note: when using WCACTION 6, single clicks are not reported until the system double click timeout has occurred (typically a .6 second delay). A similar delay will occur in list boxes when a user selects a new row with the arrow keys. The double click timing can be changed by setting the following DWORD value in the registry to the number of milliseconds to wait for the second click:

```
HKEY_CURRENT_USER\Software\DynamicConcepts\dL4\WinTerm\DoubleClickMS
```

This value will be used by all windows on the system after dL4 is started.

- o The Windows Page Printer driver and the "Preferences -> Aux Printer" dialog now support a "EOLToCRLF" option to convert either carriage return or line feed to a two character carriage return line feed sequence.
- o Enhancement: the comma alignment and TAB mnemonic sequences are now supported when the Windows printer driver is used in binary mode.
- o Bug fixed: 'WCTIPTTEXT' did not work with 'WCEDITDROP', 'WCLISTDROP', and 'WCEDITLIST' GUI elements.

Jun 28, 2011 (Maintenance Release 9.4.2)

- o Bug fixed: disabling the Preferences menu on the menu bar disabled all Preferences options in the system menu.

Apr 15, 2011 (Maintenance Release 9.4.1)

- o Bug fixed: dL4Term sometimes failed to connect when started from a short cut or command line that specified a system or script as a command line argument.

Mar 23, 2011 (Release 9.4)

- o A new mnemonic, 'SCRIPT', has been implemented to display strings that have been defined in a resource file. When using dL4Term, script strings can speed the display of complex GUI screens because scripts are cached on the client system and thus only needed to be transmitted the first time the script is used. Script strings are defined in the "[Scripts]" section of a resource file using syntax similar to dL4 string expressions. The following resource file defines a script called "Page1" for an input screen:

```
[dL4 Resources]
[Scripts]
Page1='4GRIDFONT'
; Initialize the menus
+= PChr$(101,"&Menu1","")+ 'WCMENU'
+= PChr$(102,"&Help","")+ 'WCMENUACTION'
+= PChr$(103,"&Bye","")+ 'WCMENUACTION'
+= 'WCENDMENU'
+= PChr$(102,'F4')+ 'WCACTION'
+= PChr$(103,'F3')+ 'WCACTION'
; Display the form title in double high characters
```

```

+= @120,0;+'8FONTSIZE'+ "ORDER ENTRY"+'4FONTSIZE'
; Display some labels
+= @0,8;+"Name:"+@0,16;+"Address:"+@0,24;+"City:"
+= @200,24;+"State:"+@0,32;+"ZIP:"
; Create buttons and edit boxes for input
+= PChr$(2,4,80,36,85,"OK",2)+'WCBUTTON'
+= PChr$(3,144,80,176,85,"Cancel",2)+'WCBUTTON'
+= PChr$(4,284,80,316,85,"Help",2)+'WCBUTTON'
+= PChr$(11,40,8,160,13,"",2)+'WCSTRING'
+= PChr$(12,40,16,160,21,"",2)+'WCSTRING'
+= PChr$(13,40,24,160,29,"",2)+'WCSTRING'
+= PChr$(14,240,24,280,68,"",2)+'WCLISTDROP'
+= PChr$(15,40,32,80,37,"",2)+'WCSTRING'

```

The lines beginning with "+=" are continuation lines which append characters to the current, "Page1", definition line. The space after the "+=" is not required.

The following program displays "page1":

```

Declare Intrinsic Sub SetResource
Call SetResource("example.res")
Print 'XX';PChr$("Page1");'SCRIPT';

```

Note that the resource source file shown above must be compiled into "example.res" using the MAKERES utility:

```

makeres -o example.res example.src

```

- o The resource file mechanism can now be used to update files in application defined directories. For example, a Java class file on a client system could be updated for use with a SYSTEM 31 statement. The files to be updated are placed in the "[Files]" section of a resource file with the resource name set to a logical destination directory name followed by a slash ("/") and the destination file name. The registry string value

```

HKEY_CURRENT_USER\Software\DynamicConcepts\dl4Term\WinTerm\ResourceLUMAP

```

defines the translation of logical directories to physical directories using the same syntax as an LUMAP value ("dir1=path1 dir2=path2"). Each time the application prints the mnemonic PChr\$("dir/filename")+'RESOURCE', dl4 checks if the file resource "dir/filename" has been installed and, if not, copies the file resource to the specified directory. The typical use in an application is to print one or more 'RESOURCE' mnemonic sequences and then execute a SYSTEM 31 or 33 statement that uses the files. The file resources are only copied the first time the file is used or when the resource file is updated. Please note the files will NOT be copied unless the "ResourceLUMAP" registry entry has been defined and the user has write access to the destination directories. The destination directories should not be shared between users if multiple users might be running the application at the same time.

Resource file example:

```

[dl4 Resources]
[Files]
doc/help.pdf=/usr/src/helpfile.pdf
doc/syntax.pdf=/usr/src/syntax.pdf

```

ResourceLUMAP registry value example:

```

"doc=C:\users\sam\documents lib=C:\users\sam\library"

```

Program example:

```

Print PChr$("doc/help.pdf");'RESOURCE';
Print PChr$("doc/syntax.pdf");'RESOURCE';

```

System 31,"helpviewer.exe"

- o Some applications depend on specific font or window options and will not work properly if the user changes those options. It is now possible to selectively delete any of the "Preferences" dialogs or disable any of the "Preferences->Font Size" options. To use this feature, set the registry DWORD value

HKEY_CURRENT_USER\Software\DynamicConcepts\dl4Term\WinTerm\UsePrefsMenu

to the sum of the following hexadecimal options:

1	enable all dialogs and options
2	enable "Font Size" dialog
4	enable "Window Options" dialog
8	enable "Frame" dialog
10	enable "Attributes" dialog
20	enable "Keyboard" dialog
40	enable "Translation" dialog
80	enable "Aux Printer" dialog
100	enable "Screen Print" dialog
10000	enable font name selection
20000	enable font size selection
40000	enable font color selection
80000	enable "Use Bold Font" check box
100000	enable "Use Maximum Size" check box
200000	enable "Stretch to Screen" check box
400000	enable "Menu Bar" check box
800000	enable "Edit" menu check box
1000000	enable "Prefs" menu check box
2000000	enable rows and columns boxes

For example, to enable all "Preferences" dialogs, but disallow setting the font name, the "Use Bold Font" option, the "Use Maximum Size" option, and the "Stretch to Screen" option, set "UsePrefsMenu" to the hexadecimal value C601FE (decimal 12976638).

- o A "Preferences->Translation" dialog has been added to configure translating output characters. This feature is intended for applications that depend on terminal emulators that perform similar output translation. For example, a UniBasic application might have used some 7-bit ASCII values as European characters which were not supported in UniBasic. This translation feature is intended only for printable characters. Mnemonic characters and ASCII control characters cannot be translated. Characters in GUI elements are not translated, but may be in the future. For compatibility with future releases, the translation feature should not be configured if GUI elements are used in applications.
- o The SYSTEM 31 and SYSTEM 33 statements, which execute client system commands when using dl4Term, have been extended to optionally use data files supplied from the application resource file. Resources in system commands are specified by the 'RESOURCE' mnemonic followed by a resource name. The resource name must have the same file extension needed by the command. For example, the command

SYSTEM 31,"notepad " + 'RESOURCE' + "help.txt"

runs the notepad utility on the client system using the resource "help.txt" as the text file to display. File resources for use in SYSTEM commands must be defined in the new "[Files]" section of the resource source file: Example:

```
[dl4 Resources]
[Files]
help.txt=/usr/src/helpfile.txt
```

In this release, if the system command modifies the data file, the modifications will be visible whenever a SYSTEM command uses the same

resource for the same user. This behavior may be changed in the production release of dL4.

- o A "Terminal File Printer" driver has been added to print to text files on the client system when using dL4Term.

Open #1,"filename.txt" as "Terminal File Printer"

The filename should only be a filename and must not include any path information. The user will be presented with a "SAVE" dialog to specify where the file will be stored. A "(charset=xxx)" open option can be used to construct a text file using an alternate character set (such as UTF-8).

- o The "Terminal Printer", "Default Terminal Printer", and "Windows Terminal Printer" drivers now treat the 'NR'/'WD' mnemonics as equivalent to the 'BC'/'EC' mnemonics.
- o A "-d list" command line option has been added to dl4term.exe to disable various window features and behaviors. "list" must be one or more of the keywords listed below separated by commas:

"connect" or "c" - disable the connect menu.
"preferences" or "p" - disable the preferences menu
"print" - disable the "print window" function

- o Beta bug fixed: drawing pictures and using other resource file components sometimes only worked the first time that a resource file was used.
- o Beta bug fixed: "Connect -> Clear Cache and Exit" sometimes caused a memory violation.

Oct 26, 2010 (Maintenance Release 9.3.5)

- o Bug fixed: a memory leak occurred when GUI elements were deleted after using the WCSETFONT mnemonic.

Sep 29, 2010 (Maintenance Release 9.3.4)

- o Enhancement: the "Connect" dialog host/user name field has been expanded and made horizontally scrollable to support longer values.
- o Bug fixed: if a child window extended beyond the screen, the 'NR' and 'WD' mnemonics moved the child window to an incorrect position.

Jul 27, 2010 (Maintenance Release 9.3.3)

- o Bug fixed: the 'CT' (clear tabs) mnemonic could not be used with list boxes such as 'WCLIST'.
- o Bug fixed: WCLISTGRID and WCSHOWLISTGRID list boxes did not position columns correctly. Short columns were sometimes skipped and the final column was offset significantly.
- o Bug fixed: list box thumbs were not sized correctly after an insertion.

May 28, 2010 (Maintenance Release 9.3.2)

- o Bug fixed: in list boxes, such as 'WCLIST', items could not be selected by typing the first character of the item.
- o Bug fixed: fonts sometimes appeared lighter after an 'XX' mnemonic was printed.

Mar 17, 2010 (Maintenance Release 9.3.1)

- o Bug fixed: CALL DRAWIMAGE() could not print to a Window Terminal Printer.
- o Bug fixed: the 'FITIMAGE' and 'FILLIMAGE' mnemonics could not display

files on a printer.

Mar 15, 2010 (Release 9.3)

- o Two new GUI mnemonics, 'WCLISTGRID' and 'WCSHOWLISTGRID', have been implemented to create list boxes with a grid-like appearance. The list boxes are identical to standard list boxes ('WCLIST' and 'WCSHOWLIST'), but offer these additional features:
 1. Horizontal lines are drawn between items in the list.
 2. Vertical lines are drawn at each tab position defined by the 'ST' mnemonic.
 3. The title string can include 'HT' mnemonics to align text to the grid columns.
 4. The title string can include color mnemonics.
 5. The new WCACTIONS 4 and 5 are supported to tell the application when a user has scrolled to the beginning or end of the list so that the application can add new list items.
 6. If an 'MH' mnemonic is the first character of a new item, that item will be inserted at the beginning of the list instead of the end of the list.
 7. The 'PREVPAGE' and 'NEXTPAGE' mnemonics can be used to mark special items to be displayed without grid lines. These items are placed at the beginning and end of the list to inform the user to wait while the application is reading additional items. The 'PREVPAGE' or 'NEXTPAGE' must be the first character of the special item. Any existing 'PREVPAGE' or 'NEXTPAGE' item will be replaced. These special items can be deleted from the list by printing a line that contains only a 'PREVPAGE' or 'NEXTPAGE' mnemonic.

The application must add spaces for any desired justification or centering of each field. The application must also truncate, if necessary, the field data to prevent overflowing the provided space.

Like 'WCSHOWLIST', 'WCSHOWLISTGRID' creates a display-only list box that does not allow the user to select items.

Syntax:

```
PChr$(n,x1, y1, x2, y2, "label", options);'WCLISTGRID'  
PChr$(n,x1, y1, x2, y2, "label", options);'WCSHOWLISTGRID'
```

n element number of the list box (must be unique in window)
x1,y1 Upper left grid coordinates of the rectangle containing the
 list box and title
x2,y2 Lower right grid coordinates of the rectangle enclosing the list
 box and title
label title string displayed near the list box
options an optional numeric parameter, see description below

The "options" parameter is the sum of following option values:

- 1 Disable and "gray out" the list box (user can't select the box)
- 2 Use the list box as a tab stop (in element number order)
- 4 Send input to window when value changes
- 8 Allow selection of multiple items (WCLISTGRID only)
- 16 Don't display first field of list, but return only it as the value
 (WCLISTGRID only)
- 32 Any loss of input focus is reported as an input value change
 (option 4)

The default value of "options" is zero.

As with all other GUI mnemonics, these new mnemonics can only be used with dL4Term or in dL4 for Windows.

Example program:

```

Dim Header$(100),Action$(1),Query$(2000)
Print 'XX 0 FONTCOLOR 12632256 BACKCOLOR CS';
Print PChr$(0,2^24-1);'WCSETCOLOR';
Print 'WCSETFONT';
! Create list grid element
Header$ = "Product name" + 'HT' + "Cost" + 'HT' + "Code"
Print PChr$(2,10,5,72,10,Header$,2);'WCLISTGRID';
! Place initial data in list grid
Print '2WCSELECT';
Print '20ST 30ST 40ST';
Print 'PREVPAGE GN' + "Please wait, reading preceding page."
P = 1000
N = 1019
For I = P To N
    Print "Name #";I;'HT';(I + .99) Using "#####.##";'HT';BStr$(I * 12345,16)
Next I
Print 'NEXTPAGE GN' + "Please wait, reading next page."
! Set initial position to middle of data
Print '10WCMARK';
Print '0WCSELECT';
! Set scrolling actions
Print PChr$(2,4,'F20');'WCACTION';
Print PChr$(2,5,'F21');'WCACTION';
Print PChr$(1,36,12,44,14,"OK",0);'WCBUTTON'; ! display 'ok' button
Print @10,1;"Choose a list item and press the 'OK' button."
Print 'BQ';
Do
    Input Len (1);""Action$
    If Action$ = 'F1'
        Print @0,17;"OK button pressed!"
        Exit Do
    End If
    Select Case Action$
        Case 'F20'
            ! User is at first list item, add more at the start of the list
            Print '2WCSELECT';
            For I = 1 To 10
                P = P - 1
                If P <= 900
                    ! Simulate beginning of file and delete "reading" message
                    Print 'MH PREVPAGE'
                    Exit For
                End If
                Print 'MH';
                Print "Name #";P;'HT';(P + .99) Using "#####.##";'HT';
                Print BStr$(P * 12345,16)
            Next I
            Print '0WCSELECT';
        Case 'F21'
            ! User is at last list item, add more at the end of the list
            Print '2WCSELECT';
            For I = 1 To 10
                N = N + 1
                If N >= 1100
                    ! Simulate end of file and replace "reading" message
                    Print 'NEXTPAGE RE'+ "End of list"
                    Exit For
                End If
                Print "Name #";N;'HT';(N + .99) Using "#####.##";'HT';
                Print BStr$(N * 12345,16)
            Next I
            Print '0WCSELECT';
        Case Else
            Print "Error input";BStr$(Int(Action$),16)
    End Select
Loop
Print PChr$(2);'WCQUERY';
Input ""Query$

```

```
Print 'EQ 0WCFOCUS K3';
Print @0,18;"List selection = ";Query$
End
```

- o Two new WCACTION codes have been defined for use with the 'WCLISTGRID'. 'WCSHOWLISTGRID', 'WCLIST', and 'WCSHOWLIST' mnemonics. Action 4, if defined, causes a list box to send the associated string whenever the user scrolls to the first item in the list box. Action 5, if defined, causes a list box to send the associated string whenever the user scrolls to the last item in the list box. An application can use these new actions to add additional items to a list box only when necessary.

Syntax:

```
PChr$(n,atext$);'WCACTION'
```

```
n      Element number of the input element
a      Action number to be modified:
      0      send text$ when the first list item is displayed
      1      send text$ when the last list item is displayed
text$  String value used by the action
```

- o Attribute mnemonics such as 'BU', 'EU', 'BBOLD', and 'EBOLD' can now be printed to list boxes ('WCLIST', 'WCLISTGRID', ...) as long as the mnemonics do not change the current attribute settings. For example, 'EU' can be used as long as underlining was not enabled by 'WCSETFONT'.
- o Image files for use with 'WCIMAGEBTN', 'WCDEFAULTIMGBTN', 'FITIMAGE', 'FILLIMAGE', CALL DRAWIMAGE(), and future GUI image controls can now be combined in a single resource file. Using a resource file instead of multiple image files simplifies application installation and improves performance when using dL4Term due to client side caching of resource data. To use a resource file, an application first calls the new intrinsic CALL SETRESOURCE("path") to define the current resource file. All GUI mnemonics and CALL DRAWIMAGE() will then try to load images from the resource file. If the image name can't be found in the resource file or if the name contains directory information, dL4 will try to open the name as a file. The current resource file is part of the unit context and it is passed to procedures, functions, subprograms, and through CHAIN statements. When returning from an external procedure, external function, or subprogram, the current resource file is restored to its previous value. For this reason, CALL SETRESOURCE() should be executed in the main program of an application when the application first starts. Resource files are created with the dL4 MAKERES utility.
- o A new option, "Clear Cache and Exit", has been added to the "Connect" dialog to completely clear the resource cache when dL4Term exits. This option should only be used if there is reason to believe the cache is misbehaving (for example, if image buttons are blank). The cache is not cleared if there are other dL4Term sessions running.
- o Two new mnemonics, 'WCIMAGEBTN' and 'WCDEFAULTIMGBTN', have been added to create push buttons with image files displayed on the buttons instead of title text. The mnemonic sequences to create image buttons are

```
PChr$(n,x1,y1,x2,y2,"image");'WCIMAGEBTN'
PChr$(n,x1,y1,x2,y2,"image");'WCDEFAULTIMGBTN'
```

```
PChr$(n,x1,y1,x2,y2,"image",options);'WCIMAGEBTN'
PChr$(n,x1,y1,x2,y2,"image",options);'WCDEFAULTIMGBTN'
```

```
PChr$(n,x1,y1,x2,y2,"image",options,"disabled");'WCIMAGEBTN'
PChr$(n,x1,y1,x2,y2,"image",options,"disabled");'WCDEFAULTIMGBTN'
```

where

```
n      is the GUI element number of the button
x1     is the upper left column of the button
y1     is the upper left row of the button
```


x2	is the lower right column of the button
y2	is the lower right row of the button
"image"	is the path or resource name of the image to be displayed
options	is the standard GUI element options value
"disabled"	is the path or resource name of the image to be displayed when the button is disabled. If a disabled image is not specified, a disabled image button will use an "grayed out" version of "image".

Other than using images instead of title text, image buttons operate exactly the same as text buttons.

- o Two new mnemonics, 'WCCOLORBTN' and 'WCDEFAULTCLRBTN', have been added to create push buttons that use the current GUI text and background colors. Color buttons are created and used in the same way as standard text buttons ('WCBUTTON' and 'WCDEFAULTBTN'). Examples:

```
Print PChr$(n,x1,y1,x2,y2,"title text");'WCCOLORBTN'
Print PChr$(n,x1,y1,x2,y2,"title text");'WCDEFAULTCLRBTN'
Print PChr$(n,x1,y1,x2,y2,"title text",options);'WCCOLORBTN'
Print PChr$(n,x1,y1,x2,y2,"title text",options);'WCDEFAULTCLRBTN'
```

- o Four new mnemonics, 'WCRJBUTTON', 'WCLJBUTTON', 'WCDEFAULTRJBTN', 'WCDEFAULTLJBTN', have been implemented to create buttons with right justified ("RJ") or left justified ("LJ") title text.
- o A new mnemonic, 'WCTIPTEXT', has been implemented to define tool tip text balloons that are displayed whenever a user places the pointer near a GUI element that has tip text. The mnemonic sequence

```
PChr$(n,"text");'WCTIPTEXT'
```

sets the tip text to be used for GUI element "n" in the current window. The mnemonic sequence 'N WCTIPTEXT' can be used to set the maximum width of tip text lines and thus select multiline display. Tip text is limited to a maximum of 1000 characters.

- o A new mnemonic, 'WCTITLE', has been implemented to change the title of an existing GUI control. The mnemonic sequence

```
PChr$(n,"New title");'WCTITLE'
```

is used to set the title of GUI element "n". The mnemonic sequence

```
PChr$("New title");'WCTITLE'
```

is used to set the title of the currently selected GUI element.

- o Two new mnemonics have implemented to control the size of the main window when using dL4Term or dL4 for Windows. The mnemonic 'w,h WALTSIZE' sets the main window size to "w" columns and "h" rows" as defined by the current grid coordinate setting. For example, the statement

```
Print '100,30 WALTSIZE';
```

would set the main window size to 100 columns and 30 rows. If the new window size would place the edges outside the screen, the window will be repositioned. The original window size can be restored using the 'WCSTDSIZE' or 'XX' mnemonics.

The WALTSIZE mnemonic does not change the current font size even if the new window size exceeds the screen size. The WALTSIZE and WSTDSIZE mnemonics can only be used with the main window. Windows opened on channels can be resized using the SIZE statement.

The WALTSIZE mnemonic should not be used in combination with the 'NR' or 'WD' mnemonics.

- o When using dL4Term or dL4 for Windows, new GET operation codes can be used to determine the size of a window in pixels. The new GET codes are:

```
-1042    Number of rows in pixels
-1043    Number of columns in pixels
```

The following example program prints the main window width and height in pixels:

```
Get #-3,-1042;Rows
Print "Height in pixels: ";Rows
Get #-3,-1043;Columns
Print "Width in pixels: ";Columns
```

Note: channel -3 is a special channel number for the main window. The new GET codes can be used on any channel open to a window when using dL4Term or dL4 for Windows.

- o When using dL4Term or dL4 for Windows, new GET operation codes can be used to determine the maximum possible size of a window in either grid units or pixels. The new GET codes are:

```
-1016    Maximum number of rows in grid units
-1017    Maximum number of columns in grid units
-1018    Maximum number of rows in pixels
-1019    Maximum number of columns in pixels
```

The maximum size is defined as the screen size minus space for the window frame, a menu bar, and a vertical scroll bar.

The following example program prints the maximum possible width and height in character rows and columns:

```
Get #-3,-1016;MaxRows
Print "Maximum rows: ";MaxRows
Get #-3,-1017;MaxColumns
Print "Maximum columns: ";MaxColumns
```

Note: channel -3 is a special channel number for the main window. The new GET codes can be used on any channel open to a window when using dL4Term or dL4 for Windows.

- o Two new mnemonics, 'XTDEDIT' and 'STDEDIT', have been added to enable and disable extended input action definitions in terminal definition files. When 'XTDEDIT' mode is active, any "[InputActions]" entry with the "extended" attribute will override the standard input action for a character. The extended input action definitions are normally used for the input editing as described above.
- o Two new mnemonics, 'BPSWD' and 'EPSWD', have been added to enable and disable password input mode. Using 'BPSWD' input mode, INPUT statements will echo input characters as asterisks. Example:

```
Print 'BPSWD';
Input "Password? " P$
Print 'EPSWD'
```

- o Two new open options have been added to the Window Terminal Printer driver to control duplex printing. The "DUPLEX=True" option selects duplex printing with vertical binding. The option "DUPLEXH=True" selects duplex printing aligned for horizontal binding.

```
# dl4opts=openas=Window Terminal Printer,options=duplex=true
#
# dL4 selected by dialog printer script
```

- o A new open option, "Formname=name" has been added to the Windows Terminal Printer driver to select a Windows form name. "name" must match a

Windows form name.

- o A new open option, "SourceBin=name" has been added to the Windows Page Printer driver to select a source/output bin. "name" can be either one of the standard bin names listed below or a bin code number.

Standard bin names: auto, cassette, envelope, envmanual, formsource, largecap, largefmt, lower, manual, middle, smallfmt, tractor, upper

Dec 7, 2009 (Maintenance Release 9.1.2)

- o Bug fix: proportional fonts were sized incorrectly.

Jul 6, 2009 (Maintenance Release 9.1.1)

- o Bug fix: programming a function key with the 'PGMFN' mnemonic sometimes caused non-function keys to be programmed to the same value.

May 22, 2009 (Release 9.1)

- o The 'PGMFN' and 'PGMHELPFN' mnemonics have been extended to support programming of any key. The mnemonic sequence

```
PChr$("c","text");'PGMFN'
```

will program the key that normally produces the character "c" to generate the string "text". A new check box, "Program Keys", has been added to the "Preferences -> Window Options" dialog to allow the user to disable keyboard programming. This option can be used to immediately disable programming during application development if the keyboard is programmed incorrectly.

- o A new mnemonic, 'WCCU', is supported to simplify clearing the values of a range of GUI elements. The mnemonic string '5,7WCCU' clears the GUI elements 5, 6, and 7. The upper limit of the element range can exceed the highest existing element number. The 'WCCU' mnemonic can also be used with a single parameter to clear a single GUI element.
- o The "Preferences -> Keyboard" dialog has been enhanced to allow "Unicode to Input Action" to modify parts of existing Unicode ranges instead of requiring the user to delete and then reenter the unchanged portions.
- o All "Preferences" dialogs have been changed to apply options to the current session. The new option values will take effect when the "OK" or "Apply" buttons are selected. It is no longer necessary to restart dL4 to use the new option values. The "Font Size", "Window Options", and "Frame" dialogs have "Apply" buttons which apply any changes but leave the dialog active so that changes can be tested before exiting the dialogs. Selecting the "Cancel" button does not undo changes made by the "Apply" button.
- o A "Preferences->Attributes" dialog has been added to configure how character attributes such as underline, reverse video, or protection are displayed. The dialog can be used to map any attribute or combination of attributes to a different set of attributes or colors. For example, the logical attribute of "protected" normally causes characters to be displayed as dimmed. In the "Attributes" dialog, this can be changed so that protected characters are displayed as green characters by selecting the logical attribute "Protected" in the "Current Definition", setting the color with the "Foreground" button, and then selecting the "Set" and "OK" buttons. Changing the attribute mapping effects all future output, but it does not redraw the current window or history buffer. The displayed attributes of protected characters can be changed, but protected characters always have the attribute of "protected".
- o The "Preferences->Font" dialog has been replaced with a new "Font Size" dialog. The new "Font Size" dialog has color buttons to control the default foreground and background colors. Check boxes have been added to

select a "Maximum Size" or "Stretch to Screen" window. A maximum size window uses the maximum size of the selected font that allows the window to fit on the screen. The "Stretch to Screen" window is similar to the maximum size window but may change the character aspect ratio so that the window is larger (note that "stretching" will have no effect if there is insufficient space to add one pixel to each character or if the font does not support changing the aspect ratio).

Changes in the font, font color, font size, or window size are applied to the main window when the "OK" or "Apply" buttons are selected. Only the window text is redrawn: all non-character graphics and pictures will disappear. All graphical elements such as buttons, edit boxes, or list boxes will be unchanged. Due to these limitations, it is best to apply font changes before or after running applications. Selecting the "Cancel" button does not undo changes made by the "Apply" button.

- o The "Preferences->Window" dialog has been renamed as "Window Options". All options in the dialog are applied immediately to the current session when the "OK" or "Apply" buttons are selected. The foreground and background color buttons have been moved to the new "Font Size" dialog. The "CL/CE History" check box controls whether text deleted by the 'CL' or 'CE' mnemonics is added to the history buffer (if the history buffer itself is enabled). The "Shade History" check box controls whether text in the history buffer is displayed using the original colors or with inverted colors. Dimming of protected characters is now configured in the "Preferences->Attributes" dialog and so the "No Dim BP" check box has been removed.
- o The "Preferences->Frame" dialog has been modified to remove the "Menu Bar" check box. The "Menu Bar" check box is now displayed only in the "Window Options" dialog. All options in the dialog are applied immediately to the current session when the "OK" or "Apply" buttons are selected.
- o A "Mute" item has been added to the "Edit" menu to disable the terminal bell. Selecting the "Mute" item toggles the state of the mute switch. If a check mark is displayed, the bell is muted.
- o A "/nofont" command line option has been added to the dL4Term installer to disable installation of the "Arial Mono for dL4" font. A "/noregfont" command line option has also been added to copy the "Arial Mono for dL4" files into the proper directories but skip the font registration. One of these options can be used if the dL4Term installation hangs during font installation. If the "/noregfont" option is used, the "Arial Mono for dL4" fonts can be manually installed using the installed font files.
- o Beta behavior change: the "Preferences -> Font Size" dialog no longer changes all of the text to the current foreground and background colors when a font change is applied or saved. Text colors are now changed only in text that used the previous colors and only if the foreground or background colors have been updated.
- o Beta bug fixed: the APPLY buttons in the "Preferences" dialogs updated the running session but did not save the new options in the registry.
- o Beta bug fixed: the "Preferences -> Font Size" dialog did not display font names correctly on Windows 98/ME systems.
- o Bug fixed: fonts were not displayed correctly on some systems such as Windows 98 systems with only 16 color displays.

Jul 11, 2008 (Maintenance Release 7.2.1)

- o The auxiliary printer, "Terminal Printer", and "Windows Terminal Printer" drivers now support the 'LITNUL' and 'LITCR' mnemonics on printers that have the "Binary=True" option. The 'LITNUL' mnemonic is used to output a binary zero character to a printer. The 'LITCR' mnemonic is used to output a carriage return without a linefeed to a printer that has the "CRTtoCRNL" option enabled.

- o The "Terminal Printer" and "Window Terminal Printer" drivers now support writing binary variables ("B?") to printers that have the "Binary=True" option enabled.

Jun 4, 2008 (Release 7.2)

- o The Windows Terminal driver now implements an additional window style, "HISTORY". The "HISTORY" style creates a window that has a vertical scroll bar and a text history buffer. The initial size of history buffer is based on the number of main window history pages, but the size can be increased using the 'WHISTORY' mnemonic.

Example:

```
OPEN #1,{"New Account","TITL,HISTORY",40,10} As "Window"
```

- o A new mnemonic, 'WHISTORY', has been defined to change the size of the history buffer used by a window that has the "HISTORY" style (see above). The mnemonic syntax is 'n WHISTORY' where "n" is the minimum number of history lines to be provided by the window. The size of the history buffer can only be increased; it can not be decreased or disabled.
- o When used with dL4 7.3 or later, dL4Term supports a history buffer in the dL4 debugger window.
- o A new mnemonic, 'SENDCLIP', has been implemented to cause dL4Term to send any text in the Windows clipboard as current input characters. The 'SENDCLIP' mnemonic can also be used in a function key programming string with the 'PGMFN' mnemonic to program a function key to send the clipboard characters. When 'SENDCLIP' is used in a function key, the dL4Term "Program Paste" option must be enabled.
- o A new option, "Shade History", has been added to the "Preferences->Window" dialog to cause history text to displayed using inverted colors.
- o A new option, "CL/CE History", has been added to the "Preferences->Window" dialog to cause text deleted by the 'CL' or 'CE' mnemonics to be added the history buffer. Normally, text is only added when the window is scrolled or the entire window is cleared.
- o Bug fixed: in 'WCBQRYBUF' mode, pasted characters were appended to the query queue rather than the event queue.
- o Bug fixed: the 'CE' mnemonic cleared the entire line containing the cursor even if the cursor wasn't positioned at column zero.
- o Bug fixed: the status line, when enabled, was sometimes scrolled above the bottom of the window if the user scrolled text.

Feb 7, 2008 (Release 7.1)

- o A UTF-8 character set option has been added to the SSH protocol options dialog. This option can be used on any system that uses the UTF-8 character set for SSH or telnet connections.
- o Enhancement: mouse wheels can now be used to change the window scrolling position.
- o Behavior change: copying and pasting a word includes only one trailing space.
- o Bug fixed: the printer driver "CRToCRLF" option did not work when the printer was in binary mode. Note that any current applications that used the "CRToCRLF" option and worked correctly will probably need the option turned off. This is also true for any use of the auxiliary printer in binary mode ("Binary=True,CRToCRLF=True").

- o Bug fixed: copy and paste of text from a window was corrupted if an output region was scrolled when the region was smaller than the canvas.

May 24, 2007 (Maintenance Release 6.3.3)

- o dL4Term now supports additional SSH authentication methods requiring keyboard interaction.
- o A warning and confirmation dialog is now displayed while connecting if a previously used server presents new key values. Server key values are often changed if a server is upgraded or reinstalled. Earlier revisions of dL4Term refused to connect to the server and required a manual deletion of the server identification entry from the registry.
- o Bug fixed: copy selections that included text outside the currently displayed window only copied the visible text.
- o Bug fixed: a memory violation occurred under unusual conditions while using GUI elements.

Apr 4, 2007 (Maintenance Release 6.3.2)

- o The Arial Mono for dL4 font is now included with the standard dL4Term distribution. The font has been improved to include a "Euro" currency character and a dotted zero character. The dotted zero makes it easier to distinguish a zero digit from a letter "o".

Dec 20, 2006 (Maintenance Release 6.3.1)

- o Bug fixed: the original default button status, when defined, was not restored when focus was moved to a non-button GUI element.

Oct 20, 2006 (Release 6.3)

- o dL4Term now supports secure login and encrypted communication using the SSH protocol. Both the SSH2 and SSH1 protocols are supported. To make an SSH rather than a telnet connection, simply specify the host system name as "user@system" where "user" is the user login name and "system" is the system name or IP address. dL4Term supports SSH logins via passwords or public key id files.

To use public key id files, the public key and private key files must be generated on the host system using ssh-keygen (or an equivalent utility). Please read the host system documentation for instructions on using the ssh-keygen utility and installing the public key on the host system. The private key file (in an OpenSSH compatible text format) must then be copied to a secure directory on the Windows client system (this can be done with a dL4Term SSH-by-password login followed by a cut-and-paste operation to "notepad"). The "Connect->SSH Options" dialog is used to specify the path of the private key file on the Windows system. A key file path can also be specified in a script for use by that script instead of the standard file path. The private key file passphrase, if any, will be stored in memory for the duration of the dL4Term session and reused by any later SSH connections (this can be disabled in the "Connect->SSH Options" dialog).

Warning: the private key file should be stored securely. The file should be readable only by the owner. If possible, the file should be placed in an encrypted directory. Complex passphrases are highly recommended.

If desired, dL4Term can be started in SSH only mode with telnet disabled by using the command line option "-s". A separate "dl4term SSH" link is provided under "Start->Programs->dL4Term" to run dL4Term in SSH only mode.

- o The option "SSH2 Keepalive" in the "Connect->SSH Options" dialog, when enabled, detects a failed SSH2 server or connection. The option can also be used to keep a connection from being terminated due to excessive idle

time. This option has no effect when using an SSH1 or Telnet connection.

- o Two new mnemonics have been implemented for use in GUI programs. The 'BQ' mnemonic enables a special query mode which disables echo and the cursor. The 'EQ' mnemonic disables the special query mode. Using 'BQ' instead of 'IOEE K0' can improve query performance on some systems.
- o Installation of dL4Term on 64-bit (x86) Windows is now supported. Previous versions of dL4Term ran on 64-bit Windows, but had to be installed manually.
- o Behavior change: non-ASCII punctuation characters such as true hyphens or opening/closing quotation marks are now converted to ASCII dashes or quotation marks when performing a 'WCQUERY'. The non-ASCII characters, usually generated by cut-and-paste operations from other applications, caused illegal characters errors when written to files and were not recognized by most dL4 programs.
- o Bug fixed: an address violation occurred if an entire independent window was selected with the mouse/pointer and then copied. Pasting an empty copy buffer sometimes produced an unpredictable character.
- o Bug fixed: all lines returned by querying a WCMEMO box had a space appended.
- o Beta bug fixed: input characters were occasionally lost.
- o Beta bug fixed: a memory violation sometimes occurred if the initial SSH connection attempt failed or if it was cancelled.
- o Beta bug fixed: dL4Term scripts created with older versions of dL4Term could only connect using the SSH2 protocol.

May 30 2006 (Maintenance Release 6.2.4)

- o Behavior change: horizontal tab characters printed in a window now scroll to the next line if the end of line is reached and scrolling is enabled. In previous versions, the cursor position would wrap to the top of the window.
- o A new option, "Block Cursor" has been added to the "Preferences->Window" dialog to force the cursor to be displayed as a block rather than as an underline. This option does not change the cursor appearance in GUI elements.
- o A new option, "Alt Bell", has been added to the "Preferences->Window" dialog to select an alternate sound for the "bell".
- o Bug fixed: the default button did not change when a user tabbed to a non-default push button.
- o Bug fixed: users could not tab out of a WCMEMO or WCTEXT box.

May 4 2006 (Maintenance Release 6.2.3)

- o Any value changes in a WCSTRING, WCNUMBER, or WCPRIVATE edit box can now be tracked by using the 'WCACTION' mnemonic to set the value tracking action string as action number 3. Example:

```
PRINT PCHR$(10,3,'F40');'WCACTION'; ! Set action 3 for element 10
```

Once action 3 has been defined, any change in the edit box value will be immediately reported by sending the action string ('F40' in the example) followed by the new value. For example, if the edit box had the value "AB" and the user added a "C", then the program would receive the string 'F40' + "ABC" as input. Tracking can be disabled by setting the action string to "".

- o A new mnemonic, 'AUTOCOMPLETE', has been implemented for use with WCSTRING or WCNUMBER edit boxes. The mnemonic has two parameters: the GUI element number and the autocompletion string. When processed, the 'AUTOCOMPLETE' mnemonic compares the current value of the edit box to the autocompletion string. If the current value matches (ignoring case) the leading characters of the autocompletion string, the edit box value will be replaced by the autocompletion string leaving the caret position unchanged.

The 'AUTOCOMPLETE' mnemonic is intended to be used with edit box value tracking (see WCACTION 3 above). A typical application would enable value tracking and then, each time a new tracking value was received, the application would send an 'AUTOCOMPLETE' mnemonic with a suggested value.

Example:

```
PRINT PCHR$(10,"Wilson");'AUTOCOMPLETE';
```

- o A tuning option has been implemented to redefine a "left mouse click" to immediately send the word underneath the pointer as input along with a carriage return or just a carriage return if a blank area has been selected. This option is enabled by adding 12288 (hexadecimal 1800) to the optional registry DWORD value

```
HKEY_CURRENT_USER\Software\DynamicConcepts\DL4Term\WinTerm\TuningOptions
```

or

```
HKEY_LOCAL_MACHINE\Software\DynamicConcepts\DL4Term\WinTerm\TuningOptions
```

If this option is enabled, the standard left click actions to select text will be disabled.

- o A new option, "Copy and Paste Word", has been added to the "right click" floating edit menu. If selected, the current word underneath the pointer will be copied and pasted.
- o A new mnemonic, 'SETTINGS', has been implemented in the Windows printer driver to override the printer page information supplied by Windows. The mnemonic sequence 'wmm,hmm SETTINGS' will set the usable page size to 'wmm' millimeters wide and 'hmm' millimeters high. The mnemonic sequence 'xpi,ypi,poffx,poffy,wp,hp,wmm,hmm SETTINGS' will set the following values:

```
xpi   - pixels per horizontal inch
ypi   - pixels per vertical inch
poffx - left offset in pixels
poffy - top offset in pixels
wp    - width in pixels
hp    - height in pixels
wmm   - width in millimeters
hmm   - height in millimeters
```

These values are defined to match the native Windows values. This mnemonic should not be needed in normal printer use, but it can be used if the printer driver pages incorrectly.

The 'SETTINGS' mnemonic can also be used with a single string parameter "test" to print 150 diagnostic lines which display various printer

- o Bug fixed: a memory violation could occur when changing the colors used by a GUI element.

Jan 12 2006 (Maintenance Release 6.2.2)

- o Bug fixed: the backspace key didn't work in WCNUMBER edit boxes.

Dec 27 2005 (Maintenance Release 6.2.1)

- o Behavior change: an empty menu bar is displayed if the menu bar is enabled even if all of the standard menu items are disabled. This avoids some visual problems if an application uses and updates the menu bar.
- o Bug fixed: menu bars were displayed on all windows if the menu bar was enabled on the main window.

Dec 8 2005 (Release 6.2)

- o A new GUI mnemonic, 'WCPAD', has been implemented to create transparent buttons. The WCPAD GUI element is not visible and any text or images printed underneath the button area will be visible. A left mouse click or left mouse double click anywhere in the specified area will send the associated event string (usually a function key character) as input. The behavior and usage of WCPAD is almost identical to WCBUTTON except that the button is transparent, the button cannot be used as a default, and the button does not react to the ENTER or TAB keys. If the element is queried (WCQUERY), the position of the last "click" within the button will be returned in window coordinates relative to the upper left corner of the button. The coordinates will be sent as two decimal integer strings each followed by a carriage return with the horizontal coordinate sent first. The mnemonic syntax is

```
'n,x1,y1,x2,y2 WCPAD'
```

or

```
PChr$(n,x1,y1,x2,y2,"",0,s);'WCPAD'
```

where "n" is the element number, "x1", "y1", "x2", and "y2" are numeric values specifying the upper left and lower right corners of the button rectangle, and "s" defines a scaling value. If "s" is 10, the coordinates returned by WCQUERY will be multiplied by 10 and the units will be tenths of a column or row. The default scaling value is 1.

- o A new GUI mnemonic, 'WCSHOWLIST', has been implemented to create read only list boxes. The GUI elements created by the mnemonic are identical to those created by the 'WCLIST' mnemonic except that the user is not allowed to select items and the element value cannot be queried.
- o Double clicking on the main window now selects the word underneath the pointer.
- o A tuning option has been implemented to redefine a "left mouse click" to immediately send the word underneath the pointer as input along with a carriage return. This option is enabled by adding 2048 (hexadecimal 800) to the optional registry DWORD value

```
HKEY_CURRENT_USER\Software\DynamicConcepts\dl4Term\WinTerm\TuningOptions
```

or

```
HKEY_LOCAL_MACHINE\Software\DynamicConcepts\dl4Term\WinTerm\TuningOptions
```

If this option is enabled, the standard left click actions to select text will be disabled.

- o The "Preferences->Windows" dialog has been extended to support disabling the "Connect", "Edit" and "Preferences" menus individually. Using this feature, it is possible to move the "Connect" and "Preferences" menus to the less visible system menu while leaving the "Edit" menu on the normal menu bar.
- o A new connection script item, "Message box text if failed:", has been added to allow a script to change the message text displayed when a

connection or login fails. The item must be placed before the final "Wait . . ." item. The text to be displayed must be entered in the "Text" input box before adding the message box test item.

- o Behavior change: the main window is now enlarged when a menu bar is created if the standard menu bar is disabled at startup. Previously, the window size was left unchanged and scroll bars were displayed. The window is not enlarged if the scroll bars were already present.
- o Behavior change: when using a WCNUMBER box, illegal input characters are completely ignored and will not delete existing data if the BEGIN mnemonic is used.
- o Behavior change: when GUI elements are disabled, any "marked" items will be displayed using the WCMARKCOLOR background color (if any).
- o Bug fixed: double clicking an already selected item in a WCLIST box generated a double click event but a subsequent WCQUERY to the box did not return the selected item.
- o Bug fixed: a required DLL, gdiplus.dll, wasn't always installed when needed on Windows 2000 systems.
- o Bug fix: scrolling backwards into the history buffer moved GUI elements off of the current window, but scrolling forwards did not restore the GUI elements.
- o Bug fix: the socket driver waited forever if a bad address was supplied and the OPENTIME timeout option wasn't specified. The driver will now timeout after 60 seconds.

Apr 20 2005 (Release 6.1)

- o Support has been added for the new dL4 intrinsic CALL DRAWIMAGE() which displays JPEG, BMP, and other image files in a window or to a printer. The new CALL has the following syntax:

```
Call DrawImage(Filename$, X1, Y1, X2, Y2)
```

or

```
Call DrawImage(Channel, Filename$, X1, Y1, X2, Y2)
```

where:

"Filename\$" is the path of an image file,
"X1" and "Y1" are the horizontal and vertical coordinates of the upper left corner of the rectangle in which the image should be displayed,
"X2" and "Y2" are the horizontal and vertical coordinates of the lower right corner of the rectangle in which the image should be displayed.

The image will be displayed as large as possible within the display rectangle without changing the aspect ratio of the image.

- o Two new mnemonics, 'FITIMAGE' and 'FILLIMAGE', has been defined to draw client system JPEG, BMP, and other image files in a window or on a printer. The 'FITIMAGE' mnemonic draws an image as large as possible in a specified rectangle without changing the aspect ration of the image. The 'FILLIMAGE' mnemonic stretches or shrink the image as necessary to fill the entire specified rectangle. Examples:

```
Print #1;PChr$("filepath",X1,Y1,X2,Y2);'FITIMAGE'  
Print #1;PChr$("filepath",X1,Y1,X2,Y2);'FILLIMAGE'
```

where "X1" and "Y1" are the horizontal and vertical coordinates of the upper left corner of the display rectangle and "X2" and "Y2" are the

coordinates of the lower right corner.

- o A new mnemonic, 'WCEXTKEYS', has been defined to enable extended key functionality in GUI programs. After printing the mnemonic 'WCEXTKEYS' to a dL4 or dL4Term window, pressing the ENTER key will cause the input focus to move from the current GUI element to the next GUI element with a tab stop defined (this is identical to the behavior of the TAB key). The 'WCEXTKEYS' mnemonic will also cause the ENTER key to generate a new line when typed in a WCMEMO box. These two features can be enabled or disabled individually by using a numeric parameter to the 'WCEXTKEYS' mnemonic as shown below:

```
'WCEXTKEYS' - enables both ENTER-as-TAB-between-GUI-elements
              and ENTER-as-newline in WCMEMO box.
'1WCEXTKEYS' - enables only ENTER-as-TAB-between-GUI-elements.
'2WCEXTKEYS' - enables only ENTER-as-newline in WCMEMO box.
'3WCEXTKEYS' - enables both features.
'0WCEXTKEYS' - disables both features.
```

- o The 'WCTEXT' mnemonic now supports an eighth parameter to define the logical width of the text box and enable a horizontal scroll bar. This feature is used to display text which is wider than the text box. The following example defines a text box which is 40 character wide in the window, but contains up to 80 character wide lines.

```
Print PChr$(10,20,5,60,9,"Label",2,80);'WCTEXT';
```

- o The window driver has been enhanced to support a history scrollbar feature. The new feature is disabled by default, but can be enabled by setting the "History Pages" value in the "Preferences->Window" dialog. After settings the "History Pages" value, dL4 must be restarted to use the new value.
- o The range of GUI element numbers has been increased to 1 through 9999. The previous range was 1 to 1000.
- o A new Edit menu item, "Copy And Paste", has been added to copy and paste selected text in a single action.
- o Beta bug fixed: the printer driver expected an incorrect parameter order for the 'FITIMAGE' and 'FILLIMAGE' mnemonics.
- o Bug fixed: screen corruption could occur when using some video adapters.
- o Bug fixed: the 'WCSETCOLOR' and 'WCMARKCOLOR' mnemonics did not support the special negative color values used to select window scheme colors.
- o Bug fixed: sending a 'WCQUERY' to a 'WCLISTDROP' box that had been 'WCMARK'd before initializing the box values caused a memory violation.
- o Bug fixed: GUI elements were not drawn correctly when disabled under some conditions such as program specified colors.

Jun 23 2004 (Maintenance Release 5.3.3)

- o Support has been added for the dL4 "Terminal Printer" and "Default Terminal Printer" drivers which allow applications to open program specified printers. When using these drivers, a dialog will ask the user for permission to open the specified printer. This dialog can be disabled by setting the "Open local printers" option in a connection script or by setting the registry DWORD value "AllowRemotePrint" to 1 in the registry keys

```
HKEY_CURRENT_USER\Software\DynamicConcepts\DL4Term\WinTerm\
```

or

```
HKEY_LOCAL_MACHINE\Software\DynamicConcepts\DL4Term\WinTerm\
```

- o A new option, "Run local commands", has been added to connection scripts to enable execution of commands sent by applications without asking the user to confirm permission.

May 21 2004 (Maintenance Release 5.3.2)

- o Bug fixed: binary output to the auxiliary printer occasionally had extra null characters inserted.

Apr 14 2004 (Release 5.3)

- o A new GUI mnemonic, 'FRAME', has been defined for use with dL4Term or dL4 for Windows. The new mnemonic draws a frame around a specified rectangle. The mnemonic has two formats:

```
PChr$(Left,Top,Right,Bottom);'FRAME';
PChr$(Left,Top,Right,Bottom,Type$);'FRAME';
```

The optional "Type\$" parameter specifies the frame style which can be either "Sunken" ("S"), "Raised" ("R"), "Etched" (E), or "Bump" ("B"). If the "Type\$" parameter is omitted or is "", the frame style will be similar to that of a 'WCSTRING' input box. When using the 'FRAME' mnemonic, sufficient space must be left around the rectangle to draw the frame. The frame is not a GUI element; it is drawn graphically like the 'RECT' mnemonic and the frame can be overwritten by characters or graphics. The frame color is determined by the current Windows color scheme and is not controlled by any of the color mnemonics.

- o Four new GUI mnemonics, 'WCMMSGASK', 'WCMMSGERROR', 'WCMSGINFO', and 'WCMMSGWARN', have been implemented to display standard Windows message dialog boxes. All of the mnemonics have three formats as shown below for the 'WCMMSGASK' mnemonic:

```
PChr$("Message");'WCMMSGASK';
PChr$("Message","Title");'WCMMSGASK';
PChr$("Message","Title","Options");'WCMMSGASK';
```

The "Options" string determines the number and labeling of the buttons within the message box:

"ARI"	"Abort", "Retry", "Ignore"
"O"	"Ok"
"OC"	"Ok", "Cancel"
"RC"	"Retry", "Cancel"
"YN"	"Yes", "No"
"YNC"	"Yes", "No", "Cancel"

The characters in the "Options" string may be in lower or upper case, but the first uppercase letter will select the associated button as the default button. The message box returns as input the uppercase label string of the button selected by the user followed by a carriage return. Example:

```
Print PChr$("Continue?", "I/O Error", "yN");'WCMMSGERROR';
Print 'IOEE K0';
Input "";I$
Select Case I$
    Case "YES"
        .
        .
    Case "NO"
        .
        .
End Select
```

- o A new open option has been added to the Windows Page Printer driver. The "wrap=<boolean>" option enables or disables line wrapping for

lines that exceed the maximum line length. The default is to disable line wrapping and truncate long lines. Example:

- o Behavior change: the SYSTEM 31 statement now searches additional directories when attempting to execute an operating system command. For example, the SYSTEM statement can now execute "iexplore.exe" (Internet Explorer) without an absolute path to the executable.
- o Behavior change: WCNUMBER boxes now accept commas as valid input characters. If the WCNUMBER box value is queried, the commas will not be returned as part of the value.
- o Behavior change: WCTEXT boxes no longer display scroll bars if the number of lines is exactly equal to the size of the text box.

Dec 5 2003 (Maintenance Release 5.2.4)

- o Bug fixed: GUI element labels and list box items were not sized correctly if the GUI element font size differed from the standard dialog font size.
- o Bug fixed: if the canvas was taller than the window, the display was corrupted and the scroll bar did not work.
- o Bug fixed: if the main window was minimized and then restored, scroll bars were displayed.
- o Bug fixed: a scroll bar was displayed in a GUI list box if the number of elements equaled the maximum number that could be displayed in the box.

Aug 12 2003 (Maintenance Release 5.2.3)

- o Bug fixed: the scroll bars did not work when the window was smaller than the canvas.
- o Bug fixed: memory violations occurred with some combinations of fonts and font sizes.

Jul 21 2003 (Maintenance Release 5.2.2)

- o Behavior change: copy/paste operations now supply spaces between pasted characters when necessary.
- o Bug fixed: some rare situations would cause the main window to freeze.
- o Bug fixed: input did not work with bidirectional auxiliary devices.
- o Bug fixed: the edit menu was sometimes printed as part of a print screen operation.

May 14 2003 (Maintenance Release 5.2.1)

- o A new printer type, "Bidirectional Device", is now supported as an auxiliary printer in the "Preferences->Aux Printer" dialog. When enabled, this printer type allows both writing to and reading from a device such as a serial port.
- o The raw file driver now supports a CRTOCRLF open option to append a line feed character when writing carriage return characters. This option is also available for the auxiliary printer and can be enabled in "Preferences->Aux Printer->Options".
- o Behavior change: flicker is reduced, under some conditions, when adding text to a text or edit box.
- o Bug fix: the 'PGMFN' mnemonic could not program a function key to return an ESCAPE character.
- o Bug fix: ESCAPE characters could not be sent to the auxiliary printer.

- o Bug fix: the display window was sometimes displayed in an incorrect position in the base window after the window had been maximized and the 'NR'/'WD' mnemonics had been used.

Apr 7 2003 (Release 5.2)

- o A new GUI mnemonic, 'WCMARKCOLOR', has been defined to set the text and background colors for items that have been selected by the user in GUI list boxes ('WCLIST', 'WCEDITLIST', ...). The colors are also used for items selected by the 'WCMARK' mnemonic. If the 'WCMARKCOLOR' mnemonic is printed to a window, all GUI list elements subsequently created in that window will use the specified colors. The 'WCMARKCOLOR' mnemonic can be printed to a GUI element box to set or change selection colors in only that GUI element. The 'WCMARKCOLOR' mnemonic has three formats:

```
'WCMARKCOLOR'      - use the current window text and background
                    colors
'fg WCMARKCOLOR'    - use "fg" as the text color and the current
                    window background color
'fg,bg WCMARKCOLOR' - use "fg" as the text color and "bg" as the
                    background color.
```

The color values "fg" and "bg" are numbers between -6 and 2^24 as used in the 'FONTCOLOR' and 'BACKCOLOR' mnemonics. The 'WCRESETCOLOR' mnemonic can be used to restore the standard selection colors. As with all other GUI mnemonics, this mnemonic can only be used with dL4Term or in dL4 for Windows.

- o Two new GUI mnemonics, 'WCSETCOLOR' and 'WCRESETCOLOR', have been defined to set the text and background colors in newly created GUI elements. The 'WCSETCOLOR' mnemonic copies the current window foreground and background colors. Alternatively, the 'fg bg WCSETCOLOR' mnemonic uses two numeric parameters, the text and background colors. The 'WCRESETCOLOR' mnemonic restores the standard system colors. The 'WCSETCOLOR' and 'WCRESETCOLOR' mnemonics can be printed directly to a GUI element to immediately change the text and background colors for that GUI element.
- o Color can now be controlled on a character by character basis within the list boxes of 'WCLIST', 'WCEDITLIST', 'WCDROPLIST', and 'WCEDITDROP' GUI elements. The text and background colors are set using the normal 'FONTCOLOR' and 'BACKCOLOR' mnemonics.
- o Four new standard color numbers have been defined for use with all color mnemonics such as 'FONTCOLOR'. These negative colors numbers are defined as follows:

```
-3      System standard window text color
-4      System standard window background color
-5      System standard highlighted text color
-6      System standard highlighted text background color
```

- o A new mnemonic, 'n WCASKCOLOR', has been implemented to display the standard window color selection dialog and return the color selected, if any, by the user. The mnemonic requires a single numeric parameter which is the default RGB color for the dialog. The dialog returns the selected color as a decimal input string followed by a carriage return. If no color is selected, a null string will be returned followed by a carriage return.
- o Double clicking on edit box elements such as WCSTRING or WCTEXT can now be detected by using the 'WCACTION' mnemonic to set the double click action string as action number 2. Example:

```
PRINT PCHR$(10,2,'F40');'WCACTION';
```

- o Double clicking on list box elements or buttons can now be detected by using the 'WCACTION' mnemonic to set the double click action string

as action number 2. Example:

```
PRINT PCHR$(10,2,'F40');'WCACTION';
```

- o Window contents can now be printed to a user selected printer by sending a print screen mnemonic ('PS') to a window or by the user selecting the "Edit->Print Window" menu.
- o The main window can now be maximized to occupy the entire screen. The new "Preferences->Frame" dialog configures the pattern and colors used to fill the area outside the display window.
- o ANSI terminal color control sequences are now supported for compatibility with Linux systems that default to using ANSI color sequences in directory listings ("ls" utility).
- o Bug fixed: printing to the auxiliary printer to the "Windows Terminal Printer" driver sometimes produced garbled output.
- o Bug fixed: when the main window was taller than the display window, the frame around the display window was sometimes displayed incorrectly.
- o Bug fixed: if a printer script used the "TRANSLATE=" option and did not use the "PATH=" option, a "File not found" error sometimes caused a memory violation.
- o Beta bug fixed: colors were set incorrectly by 'FONTCOLOR' or 'BACKCOLOR' when used in a GUI list box such as 'WCLIST'.
- o Beta bug fixed: the final item in a GUI list box (such as 'WCLIST') was sometimes displayed with an incorrect value whenever the list box items were initialized.

Jan 6 2003 (Maintenance Release 5.1.3)

- o Bug fixed: if a WCGROUP with radio buttons was enclosed within another WCGROUP, the first button would be placed in a separate radio group apart from the other radio buttons.
- o Bug fixed: binary output did not work on Windows 95/98/ME systems if the printer used direct printing rather than spooling.
- o Bug fixed: if a printer script used the "TRANSLATE=" option and did not use the "PATH=" option, a "File not found" error sometimes caused a memory violation.

Sep 6 2002 (Maintenance Release 5.1.2)

- o Bug fixed: the 'CS' mnemonic and statements using the 'CS' mnemonic deleted GUI menu elements.

Sep 4 2002 (Maintenance Release 5.1.1)

- o A new mnemonic, 'DEFAULTCOLOR', has been defined to set the default colors to the current foreground and background colors for the rest of the session.
- o Enhancement: up to 128 GUI elements can now be queried at one time using one or more WCQUERY mnemonics (previously only 32 elements could be queried before reading the results).
- o Bug fixed: 'WCLISTDROP' elements whose drop down boxes extended below the bottom of the window could not be deleted by the 'CS' mnemonic.

Jun 3 2002 (Release 5.1)

- o The 'n,m WCENABLE' and 'n,m WCDISABLE' mnemonics can now be used to enable or disable all GUI elements between "n" and "m" inclusive.

- o A new mnemonic, 'RESETFONT', has been defined to restore the default screen or printer font name and size.
- o A horizontal scroll bar is now automatically displayed by WCLIST boxes if the list elements are too wide for the list box.
- o The Windows Page Printer driver now tries to match the requested line height even when the system printer driver cannot provide the requested font height.
- o An input mode has been added to defer keyboard input after a GUI event has been reported. This prevents the user from entering data into the wrong GUI element (such as a 'WCSTRING' edit box) because the application hasn't had time to select the proper new GUI element. The defer mode is activated by printing a '1 WCEVENT' mnemonic string. Any GUI event that returns an action string to the application will cause keyboard input to be stored in an internal queue until the application prints a 'WCFOCUS' or 'BEGIN' mnemonic to set the input focus. After the focus is set, the queued keyboard input will be released to the selected GUI element until another GUI event occurs.

Once selected by a '1 WCEVENT' mnemonic string, the deferred input mode applies to all windows and all input until a '0 WCEVENT' mnemonic string is printed. The current deferred input queue can be cleared by using a '2 WCEVENT' mnemonic string.

To prevent locking out the debugger and command mode, printing a 'K1', 'K2', or 'K3' cursor mnemonic will delete all queue input and re-enable input (input deferral mode itself is not disabled).

- o Behavior change: the maximum repeat count for a character (such as 'n PI') is now limited to 32767 to avoid runaway printing.
- o Behavior change: the 'K1', 'K2', and 'K3' mnemonics no longer clear the deferred input ('1WCEVENT') mode. If input deferred mode is used with dL4 4.4.1.4 or earlier, entering the debugger or command mode may cause a 30 second keyboard lockup because input will continue to be deferred until the timeout occurs.
- o Beta bug fixed: 'WCLIST' list boxes did not work with hidden fields or null ("") list items.
- o Bug fixed: if the connection was lost while dL4Term was minimized, dL4Term could not easily be restored or maximized.
- o Bug fixed: the characters 'SI' ("\017\") and 'FS' ("\034\") could not be printed to a dL4Term auxiliary printer.
- o Bug fixed: the "Printer->Preferences" dialog overwrote some printer parameters that are configurable only via the registry editor.
- o Bug fixed: closing a child window reported a GUI event to the parent window if the current parent window GUI element had the queue-on-loss-of-focus option set.
- o Bug fixed: functions keys were not passed from GUI elements to the parent window on Windows 95, 98, or ME.
- o Fix bug: typing a function while input was being deferred ('1WCEVENT' mode) caused an infinite loop.
- o Fix bug: printing in binary mode ("binary=true") sometimes lost data.
- o Fix bug: users could not TAB between GUI elements in a child window if input was being deferred ('1WCEVENT' mode).
- o Fix bug: output sometimes froze making it necessary for the user to

terminate the dL4Term process.

- o Fix bug: printing 'CU' to a 'WCSTRING' edit box when it had the input focus caused the edit box to be flag as "changed". A "value changed" action string was sent the edit box lost the input focus.

Dec 4 2001 (Release 4.4)

- o Support for two new dL4 mnemonics: 'WCBQRYBUF' and 'WCEQRYBUF'. Printing 'WCBQRYBUF' to a window enables special buffering for the data returned by the 'WCQUERY' mnemonic. Normally, sending 'WCQUERY' to a GUI element (such as 'WCSTRING') causes the element to return its current value. For example, sending 'n WCQUERY' where 'n' is the element number of a 'WCSTRING' edit box causes the text in the edit box to be returned as input to the window. This can cause complicate programming because query results are mixed in with event data such as function keys or GUI element status changes. The special buffering enabled by 'WCBQRYBUF' separates the query results from the event data. Event data can be read by normal INPUT statements, but the query buffer can only be read from "record 1" of the window channel. For example,

```
Dim S$(100)
Declare Intrinsic Sub DupChannel
Call DupChannel(99,-4)
Print 'WCBQRYBUF 5 WCQUERY';
Input #99,1;S$
```

Once a 'WCBQRYBUF' mnemonic is printed, the special buffering mode applies to all windows and persists until a 'WCEQRYBUF' or 'IORS' mnemonic is printed.

- o Support for a new dL4 mnemonic, 'PGMHELPPFN'. The mnemonic sequence

```
PChr$(n,"text");'PGMHELPPFN'
```

will program function key 'n' ('Fn') to send the character string "text" as input when pressed. In addition to "text", the function key will send the action string of the currently selected GUI element, if any. This mnemonic is similar to 'PGMFN', but can be used to implement context dependent function keys such as help or search keys. 'RF' or 'XX' mnemonics can be used to reset all function keys to their original values of 'Fn'.

- o Support for a new dL4 function, MSC(45), to return the GUI element number of the user selected GUI element or a WCQUERYed GUI element. The value returned is not the current element number, but rather the number at the time of the most recent input. For example, assume a user had selected edit box 1, typed 'F6', moved to edit box 5, and then typed 'F6' again. If an INPUT statement read only the first function key value, then MSC(45) would return 1, the selected element at the time the function key was pressed. The value of MSC(45) would not change to 5 until the second function key value was read by an INPUT or READ statement.
- o A new mnemonic, 'WCRESETFONT', has been defined for dL4 GUI programming. The new mnemonic restores the default font for use by newly created GUI elements. Typically, the mnemonic would be used to clear the font set by a previous 'WCSETFONT' mnemonic.
- o The Preferences->Printer dialog now allows the user to set an "Options" value. The "binary=true" option can be used with non-device printers to allow unformatted character sequences, such as PCL commands, to be sent directly to a printer (device printers always operate in a binary mode).
- o dL4Term now allows dL4 for Unix applications to execute commands on the user's PC via the SYSTEM 31 statement. A message box will display

each command before it is executed and allow the user to accept or cancel the command. The DWORD registry value

HKEY_CURRENT_USER\Software\DynamicConcepts\dl4Term\WinTerm\AllowSYSTEMCmd

or

HKEY_LOCAL_MACHINE\Software\DynamicConcepts\dl4Term\WinTerm\AllowSYSTEMCmd

can be set to one to automatically accept commands without displaying the message box (note: if set to zero, an "AllowSYSTEMCmd" value in HKEY_CURRENT_USER will require the message box no matter how the HKEY_LOCAL_MACHINE value is set). This feature requires dl4 for Unix 4.3.2.6 or later.

- o dl4Term now allows dl4 for Unix applications to open a user selected local printer on a channel. This feature requires dl4 for Unix 4.3.2.6 or later.
- o Fix bug: auxiliary printer output was double or triple spaced when dl4Term was used with UniBasic.
- o Fix bug: dl4Term sometimes "froze" after an ESCAPE or other interrupt key was pressed.

May 9 2001 (Maintenance Release 4.3.1)

- o Two new open options have been added to the Windows Page Printer driver. The "LPI=n" option selects a default font size such that "n" lines per inch will be printed. The "CPI=n" option selects a default font size such that "n" characters per inch will be printed. The two options can be used together. The following example shows a printer script that uses 8 lines per inch, 10 characters per inch, with half inch horizontal and vertical margins:

```
rem dl4opts=openas=selected page printer,lpi=8,cpi=10,hmargin=36,vmargin=36
rem
rem dl4 selected by dialog printer script
```

- o Bug fixed: multiple selection list boxes (such as 'WCLIST') did not work on Windows 95/98/ME systems if two or more items were selected.

Mar 15 2001 (Release 4.3)

- o A new mnemonic, 'WCDEFAULTBTN', has been implemented to create default push buttons. These buttons are identical to those created by the 'WCBUTTON' mnemonic except that the button is marked as the default and will be triggered by the user typing ENTER if the focus is on the button or on another non-pushbutton GUI input element. The arguments to the 'WCDEFAULTBTN' mnemonic are identical to those of the 'WCBUTTON' mnemonic. Note: if the 'nWCFOCUS' mnemonic is used to set focus to a non-default pushbutton, that button becomes the default.
- o Enhancement: the 'BEGIN' mnemonic can now be sent to GUI edit boxes (such as 'WCSTRING') to place the cursor at the start of the current text and to select the current text for replacement. The user can tab or otherwise move to a different GUI element to leave the value unchanged or type new characters to completely replace the existing value. To send the 'BEGIN' mnemonic to an edit box, the edit box must be the currently selected GUI element (via 'nWCSELECT'). The following example could be used to initialize the contents of a newly created 'WCSTRING' edit box and select the contents for replacement:

```
Print '4WCSELECT';"Initial Data";'BEGIN';'0WCSELECT';
```

The 'n BEGIN' mnemonic can be used to perform the 'BEGIN' function on GUI element "n" and also set the input focus to that element:

Print '4BEGIN';

- o New mnemonics 'n,d GRIDENGLISH', 'n,d GRIDMETRIC', and 'n,d GRIDFONT' have been created to provide a higher precision method of setting the coordinate grid. In these new two parameter mnemonics, "n" and "d" are the numerator and denominator of a fraction which is passed to the mnemonic as a parameter. Thus the mnemonic string '1000,72 GRIDENGLISH' is equivalent to 'x GRIDENGLISH' where "x" is equal to 1000/72. Since 'GRIDENGLISH' sets the coordinate grid in 1000ths of an inch, this sets the grid to 1/72 inches (points). These mnemonics are supported by both the window and page printer drivers.
- o A new mnemonic 'n LPI' has been implemented to set the number of lines per inch when printing to the Page Printer driver.
- o A new mnemonic 'n CPI' has been implemented to set the number of characters (columns) per inch when printing to the Page Printer driver. A second form, 'n,d CPI' has been implemented to use the fraction "n/d" when setting the number of columns per inch.
- o A new mnemonic, 'n FONTCELL', has been implemented to set the font height by specifying the line height, rather than the character height. The mnemonic argument is in grid coordinates and is identical to that used by 'n FONTSIZE'.
- o A new mnemonic, 'MARGIN', has been implemented to set horizontal ('w MARGIN') and vertical ('w,h MARGIN') margins in the Page Printer driver. The arguments to the mnemonic are margins expressed in grid coordinate system units. For example, if the current coordinate grid was tenth inches ('100GRIDENGLISH'), then a half inch left margin could be set by '5MARGIN' or a half inch left margin combined with a one inch top/bottom margin could be set by '5,10MARGIN'.
- o A new printer mnemonic, 'nLANDSCAPE', has been implemented to select landscape mode if "n" is 1 and portrait mode if "n" is 0. The mnemonic should only be used at the beginning of a page.
- o The GUI window driver now treats the 'LK' (lock keyboard) and 'UK' (unlock keyboard) mnemonics as no-ops rather than reporting an illegal character error.
- o Behavior change: if the input focus is on a 'WCBUTTON' button and the tab option is enabled on at least one GUI element in the same window, then typing ENTER will trigger the button.
- o Enhancement: the drop down section of drop down lists ('WCLISTDROP' and 'WCEDITDROP') can now extend below the bottom of the window on which the list is placed. This makes it possible to place a drop down list near the bottom of a window. If the drop down section extends below the bottom of the screen, the drop down section will be displayed above the normal position.
- o Enhancement: the 'WCQUERY' mnemonic can now be used with two numeric parameters. The mnemonic 'm,n WCQUERY' (or PChr\$(m,n);'WCQUERY') performs a WCQUERY on all GUI elements between m and n inclusive.
- o ESCAPE, ABORT, and function key ('Fn') characters typed in a GUI input element (such as 'WCSTRING') are now passed to the parent window of the GUI element. This allows applications to use ESCAPE and function keys to trigger special actions in GUI programs.
- o A new mnemonic, 'n WCWHERE', has been implemented to return as input the action 'n' string value of the GUI element currently selected as the input focus. See 'WCACTION' in the dL4 GUI tutorial for details on action strings.

- o "Clicking" on a selected item in a list box ('WCLIST') now unselects the item.
- o The 'WCUNMARK' mnemonic is now supported for single selection list boxes.
- o The hidden field option, option 16, is now supported in list drop boxes ('WCLISTDROP').
- o Behavior change: the MODAL style is no longer ignored in child windows. A MODAL child window will disable all GUI input elements and typing in the parent window. The parent window itself is not otherwise disabled because that would disable the child window.
- o A new option value, 32, has been implemented for all GUI input elements. The new option causes any loss of input focus to be reported as an input value change (option 2).
- o Buttons ('WCBUTTON'), check boxes ('WCCHECK'), and radio buttons ('WCRADIO') now support multiline labels. Carriage returns ('CR') can be placed in labels to explicitly split lines.
- o Behavior change: list boxes ('WCLIST') now occupy the full vertical space assigned to the box. Previously, the height was truncated to eliminate any partially display list items (a list box that was 4.5 items high was displayed as 4 items high). This change makes it possible to align the bottoms of list boxes with varying heights.
- o Behavior change: the 'BG' and 'EG' mnemonics are now ignored by the Page Printer driver instead of causing an "illegal character" error.
- o Bug fixed: closing a window and then immediately opening a new window caused some output for that new window to be directed to the main window.
- o Bug fixed: radio buttons reported a value change only on the first occurrence.
- o Bug fixed: new independent windows were sized using the coordinate system of the last accessed window instead of the main window.
- o Bug fixed: 'nWCMARK' did not work after 'CU' was sent to a list box or drop down list box. Also fixed a memory leak when using hidden fields.
- o Bug fixed: GUI option 4, report changed value, treated any gain of focus in a radio button as a changed value.
- o Bug fixed: GUI elements and child windows were not positioned correctly when a window was scrolled from the 0,0 position and under some other circumstances.
- o Bug fixed: If GUI input elements with tab options existed in the parent window, it was not possible to enter input in a child window.
- o Bug fixed: the 'n WCUNMARK' mnemonic would unselect item "m" in a single selection list box if "m" was currently selected, but item "n" was not.
- o Bug fixed: tabbing between GUI input elements is now supported in child windows and is no longer restricted to the main and independent windows.
- o Bug fixed: outputting a bad mnemonic sequence to the Page Printer driver sometimes caused the driver to hang.

Aug 1 2000 (Release 4.2)

- o The terminal definition files in the samples directory are now

supplied as Unix text files rather than DOS files.

- o Scrolling performance of a full window has been improved by 50 to 80 percent on typical systems.
- o The dL4/UniBasic 'BA', 'EA', 'BO', 'EO', 'AE', and 'AD' mnemonics are now supported to direct output to an "auxiliary printer". When printer output is enabled, output will be sent to the printer selected via the "Printer" sub-menu of the "Preferences" menu. That printer can be configured to be a specific Windows printer, the default Windows printer, a user selected Windows printer, a COMn serial device, or an LPTn printer device. When sending output to a Windows printer, a document will be considered complete and all pages will be printed when auxiliary output is disabled by an 'EA', 'EO', 'AD', or 'XX' mnemonic. A program should not expect to be able to suspend output with 'EA', 'EO', or 'AD' and then later continue output on the same printer page.
- o The dL4/UniBasic 'WS', 'ES', 'SO', and 'SF' mnemonics are now supported to display a status line the bottom of a window. The 'WS' mnemonic redirects all output to the status line until an 'ES' mnemonic is output. The 'SO' (status line on) and 'SF' (status line off) mnemonics control whether the status line is visible. Note: the mnemonics and status line text must be output to the window that will contain the status line. When using dynamic windows ("WINDOW OPEN"), the "WINDOW OFF" statement or the CALL DUPCHANNEL intrinsic should be used to output status line mnemonics to the main window rather than the current window.
- o The dL4/UniBasic 'WD' and 'NR' mnemonics have been implemented to change the current font size to either the default font size or 6 tenths of the default font size. Assuming a default window width of 80 characters, the 'NR' mnemonic will set the font and window sizes to provide a width of 132 narrow characters. Both mnemonics clear the window with the equivalent of a 'CS' mnemonic.
- o A new mnemonic, 'PGMFN' has been defined to control the text sent by the 'Fn' function keys. The mnemonic sequence

```
PChr$(n,"text");'PGMFN'
```

will program function key 'n' ('Fn') to send the character string "text" as input when pressed. The 'RF' or 'XX' mnemonics can be used to reset all function keys to their original values of 'Fn'. This mnemonic will be supported by dL4 4.2.

The 'PGMFN' mnemonic has the Unicode value of 0xF1BB.

- o A new mnemonic, 'INVERT' has been defined to invert colors within a specified region. The mnemonic has the following forms:

'INVERT'	- Invert colors from the current position to the end of the current line.
'n INVERT'	- Invert colors from the current position for 'n' characters.
'w,h INVERT'	- Invert colors from the current position within a rectangle of 'w' characters width and 'h' characters height.
'x1,y1,x2,y2 INVERT'	- Invert colors within rectangle defined by the points 'x1,y1' and 'x2,y2' of the current coordinate grid.

The 'INVERT' mnemonic has the Unicode value of 0xF1BA. This mnemonic will be supported by dL4 4.2.

- o In Windows, the user can exit dL4 by selecting the exit button or the "close" action of the system menu. Exiting dL4 in this manner prevents the application from cleaning up any partially written

transaction. A new mnemonic, 'ONCLOSE', has been defined to allow application to control such exits. The 'ONCLOSE' mnemonic has the following forms:

<code>PChr\$(0,"text");'ONCLOSE'</code>	- Display "text" within a message box and give the user a choice of exiting dL4 or continuing the application.
<code>PChr\$(1,"text");'ONCLOSE'</code>	- Display "text" within a message box and prevent the user from exiting dL4.
<code>PChr\$(2,"text");'ONCLOSE'</code>	- Prevent the user from exiting dL4 and treat "text" as input for the program.
<code>PChr\$(0,"");'ONCLOSE'</code>	- Disable any previously set ONCLOSE action.

Programs using the 'ONCLOSE' mnemonic should output a null action ('PChr\$(0,"");'ONCLOSE') or an 'XX' mnemonic before exiting to clear the ONCLOSE action. This mnemonic will be supported by dL4 4.2.

The 'ONCLOSE' mnemonic has the Unicode value of 0xF1BC.

- o The "Initial Values" sub-menu of the "Preferences" menu has been renamed as the "Window" sub-menu.
- o Behavior change: the 'WCNUMBER' mnemonic now produces an edit box that accepts signed floating point numbers rather than just non-negative integers. The numbers must be in the format <sign><digits><"."><digits> where the sign and decimal point are optional. Entering an illegal character will be rejected. Note: the edit box may return a return a value such as "-", ".", or "-." if the user starts to enter a legal number but never adds the expected digits.
- o Behavior change: the default keyboard configuration now defines the shifted function keys as F13 through F24. For example, pressing the "F1" key while holding down the shift key will generate an 'F13' character. This change only effects the default configuration and will not add the additional function keys to an existing keyboard configuration (set if the Preferences->Keyboard menu item has been used).
- o Beta bug fixed: the 'NR' and 'WD' mnemonics did not update the MSC(33) and MSC(40) function values (number of columns). This fix requires use of dL4 4.2 (beta 3 or later).
- o Beta bug fixed: using 'WD', 'NR', or font mnemonics sometimes caused a resource leak and eventually an error exit from scopew/runw.
- o Beta bug fixed: outputting illegal characters while the auxiliary printer was enabled sometimes caused legal characters to be lost.
- o Bug fixed: if protected characters and blinking characters were used together, the blinking area was not always cleared by the 'CS', 'XX', or other "clear" mnemonics.
- o Bug fixed: a selection in a list box or drop-down list box is reported immediately if the value changed option is used.
- o Bug fixed: GUI elements such as buttons sometimes needed to be "clicked" several times instead of just once.
- o Bug fixed: GUI elements such as buttons sometimes returned incorrect values when selected or changed.
- o Bug fixed: selecting a value from a drop list of a GUI element was not reported when requested by the WCACTION mnemonic.

- o Bug fixed: using a font size larger than the window caused a memory violation.
- o Bug fixed: typing a RETURN while a paste area was selected and a input was active caused a memory violation.
- o Bug fixed: opening a MODAL window in dL4 for Unix caused dL4Term to exit after displaying an "Unexpected error" message box.

Mar 10 2000 (Release 4.1.2)

- o Maintenance release to fix various problems (see below).
- o The 'MH' (Move Home) mnemonic is now supported in edit boxes to move the current position to the beginning of the text in the box.
- o Bug fixed: the input focus was not moved to the current window if binary or activate-on-control-character input mode was used.
- o Bug fixed: scroll bars weren't enabled or disabled at appropriate times in multiline edit boxes ('WCEDIT' and 'WCTEXT' mnemonics).

Dec 2 1999 (Release 4.1.1)

- o Maintenance release to remove beta period expiration.

Nov 17 1999 (Release 4.1)

- o First production release of dL4Term for Windows.

Notes

=====

- o dL4Term for Windows cannot be used with Windows 3.1. dL4Term will run under the Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP, Windows 2003, Windows 2008, or Windows Vista operating systems.
- o The custom font, "Arial Mono for dL4" has been licensed by DCI for delivery with dL4Term. The font is a standard windows TrueType font and is compatible with other Windows applications, including Telnet, dL4 and IQ for Windows. "Arial Mono for dL4" has been included with the dL4Term license to ensure users have access to an highly legible font on each properly licensed dL4Term installation. Do not install or use these fonts on any system which does not have a valid dL4Term license. Arial is a trademark of The Monotype Corporation registered in the U. S. Patent and Trademark Office and certain other jurisdictions.
- o To start a SSH or telnet connection select the Connect function of the "Connect" menu to display the "Connect to host" dialog. A system name or script name can be entered in the "host" box or selected from the drop down list. For SSH connections, the login name must be specified along with the system name using the syntax "user@system". When using scripts with SSH, the login name is specified in the script. The drop down list contains the most recently accessed addresses and all script names. Select the "Connect" button to proceed with the connection.
- o In the "Connect to host" dialog, the system name may either be an actual system address or a script name. A system name can include a service name or port number for connection to non-standard SSH/telnet ports or non-telnet services. The service name or port number is specified as suffix to the host name separate by a colon ("decker:daytime").
- o Shortcut icons can be created to automatically start an SSH or telnet session using dL4Term. Simply create a shortcut to dL4Term and then modify the shortcut properties to add a system or script name to the target string. If the target string is quoted, add the system or

script name after the closing quotation mark and separated by a space.

- o The Connect function cannot be used if a connection has already been established. To end the current connection, exit from the host system or use the Disconnect function of the "Connect" menu.
- o A connection can be terminated by exiting the host session (on a Unix host, type "exit" in the command shell), by using the Disconnect function of the "Connect" menu, or by exiting from dL4Term. Always exit from any host application before ending the connection so that the application can cleanly close any open files.
- o The Scripting function of the "Connect" menu makes it possible to create and modify login scripts. Login scripts are used to automate SSH or telnet sessions. A script can establish a session, log on to the host session, and then start an application. A script consists of one or more actions selected from the Action drop down box. A typical script to log into a Unix system is:

```
Start
Wait for Host: ogin:
Send User Id <CR> to Host
Wait for Host: ssword:
Send Password <CR> to Host
```

Note that the strings "ogin:" and "ssword:" are arguments to the "Wait for Host:" action and must be entered in the Text box. When the script is used, dL4Term will display a dialog box for the user to enter a user id and password. The script will then attempt to log into the system. If attempt fails, the script will timeout in 60 seconds (the "Wait failed if Host sends:" action can be used to detect a failed login without waiting for a timeout).

- o Scripts can either be shared by all users or belong only to the current user. Set or clear the "Shared Script" check box to control how a script is saved. All scripts are saved in the registry under the key Software\DynamicConcepts\dL4Term\Scripts in either HKEY_LOCAL_MACHINE (shared scripts) or HKEY_CURRENT_USER (personal scripts). Scripts can be copied between systems by exporting and then importing the script registry entry in Regedit.
- o In the Scripting dialog, the Display check box controls whether host output is displayed while the script is running. The "Exit on Disconnect" check box, if set, causes dL4Term to exit when the session ends.
- o The "Reset Terminal" function of the "Connect" menu can be used to reset the clear the screen, restore the default font, and delete all GUI objects. This function should only be used after a fatal application error leaves the screen in an unusable state.
- o Copy and Paste functions are available via a popup menu invoked by pressing the right mouse button. The "Edit" menu is also part of the main window menu bar or part of the System menu ("Alt-Space").
- o Font characteristics, window size, and keyboard translation can be modified using the "Preferences" menu. The "Preferences" menu is part of the main window menu bar or part of the System menu ("Alt-Space"). The "Preferences" menu provides access to the "Font", "Initial Values", and "Keyboard" dialogs. Features that are controlled by the "Initial Values" dialog will not change until the next entry into dL4term. Changing the font or font size will immediately effect the current font used to draw new characters, but will not change existing text or the window size.
- o The number of lines and columns in the main window is controlled by registry settings and defaults to 25 lines of 80 columns. The values cannot be changed for a running instance of dL4Term, but the registry

values can be modified by the "Initial Values" dialog for use by the next entry into dL4Term. In that dialog, the canvas values control the actual number of rows and columns while the window values control the number of rows and columns that are displayed. Note that the terminal description files used by terminfo and UniBasic assume a window size of 25 lines and 80 columns.

- o The default foreground and background colors are controlled by registry settings and default to those of the system color scheme. The default values cannot be changed for a running instance of dL4Term, but the registry values can be modified for future invocations by using the "Initial Values" dialog.
- o Keyboard translations can be modified using the "Keyboard" dialog of the "Preferences" menu. Changes are applied immediately and are saved in the registry for all future invocations of dL4Term. The "Special Key to Unicode" section is used to examine, add, or modify translations of key combinations to Unicode characters. For example, to define "Shift-F1" to be the character 'F13', enter "F14D", the Unicode character value of 'F13', in the edit box under the heading "Unicode", select the edit box under the heading "Key Combination", press the SHIFT and F1 keys, and then click "Set" and "Ok". Warning: it is possible to intercept normal Windows keys such as Alt-F4 (Close Window/Application) and thus interfere with normal Windows keyboard usage. In the list box, a "Key Combination" value of "VK=xxx" indicates a key combination which isn't supported by the keyboard.

Platforms =====

Operating system information for dL4Term 10.1.4 production:

ID	Platform	OS
1B	Win32 Intel	Windows NT/2000/XP/2003/2008/Vista/7/8