AlphaLAN^{++}

Version 8.0
(Build 4.0 onwards)

Release Notes

U. A. Systems, Inc.

TEL: (909) 608-1575
FAX: (909) 608-1578

Rev: 8040

Table of Contents

AlphaLAN++ 8.0 (build 4.0 onwards) Release Notes

Version 8.0

Date: May 18, 2002 **Rev: 8040**

Introduction

AlphaLAN++, version 8.0, Build 4.x, offers many new and advanced features over earlier builds of version 8.0 and its previous version 7.1. AlphaLAN++ is a 32-bit, multi-threaded, MDI application and is one of the fastest Terminal Emulation software available commercially.

1.0 New features (Build 4.0 onwards)

- **Host Programming Interface.** Build 4.0 allows you to enhance your Unix and AMOS applications by using a variety of new Windows Interface functions. These functions are broadly classified in the following categories.

    1. **DDE Client Interface**. This interface allows you to pass information back & forth between your host application and Windows Applications (DDE Server). The complete range of DDE functions like DDEInitialize, DDEExecute, DDEPoke, DDERequest, DDEAdvise, DDEUnadvise, and DDETerminate are available to you to make full use of DDE conversations with Windows DDE Server applications.
    2. **PC File and Clipboard Interface**. A variety of interface functions are now available to you to send (write) the host data to PC Files/Clipboard and receive(read) data from PC Files & Clipboard.
    3. **Advanced terminal emulation interface**. A variety of specialized and advanced terminal features are available to your host applications even though the actual terminal may not offer them. These include, Save/Restore Rectangular Box/Area, Alternate Page, etc.
    4. **Miscellaneous functions interface.** A variety of functions offering FTP file transfer interface, launching of Windows applications, etc. are also available.

- **Terminal Emulation of advance features in Wyse50, Wyse 60, Wyse 150 and Wyse 350 Terminals.** Build 4.0 offers the following advance functions. Program function keys and other special keys, Send Cursor Line, Back Tab, Insert Mode support, Monitor Mode Support, Horizontal Split, Lower/Raise Horizontal Split Screen.

- **Terminal Emulation of advance features in Esprit III terminal.** Build 4.0 offers the following advance functions. Program function keys and other special keys, Send Cursor Line, Back Tab, Insert Mode support, and Monitor Mode Support. You can also load a substitute character for the space character when executing Clear Screen commands.

- **Enhanced Printer Support.** Build 4.0 now supports printers that do not support “raw” mode like HP 1000, etc.

- **Force Num Lock and Caps Lock keys:** You can now setup the terminal emulator to force Num Lock, or Caps Lock, or both keys. This allows you to switch between your host application and windows applications without having to turn these keys on and off.

- **Allow multiple sessions on start up.** Build 4.0 would allow you to start multiple sessions on start up and let you Tile or Cascade these windows, automatically.

- **Save individual window sizes for multiple processes.** Build 4.0 saves the window position of the application under the configuration file name (.aln file). This would allow you to start multiple processes of the applications (using separate .aln files) and have individual window sizes and positions for each of these application processes (sessions).

- **Enhanced Rebar for UDK buttons.** A number of changes have been made to the "Rebar" that shows 4 UDK buttons (and the address bar) as follows

    1. The button width adjusts automatically now, based on the length of the text on the buttons.

    2. You can also choose to limit the text shown on the button to a specific number of characters without any need to re-enter the label text for each button.

    3. The image shown on the buttons may now be hidden to save vertical screen space.

    4. A menu item "Arrange Buttons" has been added to "refresh" positions of UDK bands and buttons.

    5. The 4 UDK Bar bands will 'maximize/minimize" on double clicking the mouse in the "level name" area of the band (instead of single click).

    6. The order of the bands within the "Rebar" is now fixed. The first band is the "Address Bar" (if shown) followed by the four UDK bars (UDK1 to UDK4). If you move one band to another row, the bands shown after the band being moved would also move. For example, if you move UDK2 bar to the second row, then UDK3 and UDK4 bar would also move to the second row to keep the same (fixed) order. This avoids display confusion where bands were getting interchanged.

    7. The UDK band headers that display the "level name" for the UDK bars will automatically adjust in width to accommodate full level name as typed.

2.0 New features (version 8.0 over version 7.1)

- **Internet Browser Support.** Version 8.0 can start an "Internet Browser Session". The browser session offers a "Navigation Toolbar" with browser functions like Stop, Refresh, Back, Forward, Close, etc. It can support multiple "Terminal" and "Browser" sessions and you can easily switch between these ""sessions". You can start a browser session by using the menu, or typing the "URL" in the Address bar, or by using a special "Escape Code" from your host program. A new command line option (/b) is available to start the application with a browser session instead of a terminal session.

- **An Address bar and URL Support.** Version 8.0 supports an address bar in each terminal session. Address bar allows you to type in any URL. AlphaLAN will start a new browser session for your URL. If you type a FTP URL, or a mail URL, it will start UA-FTP session or your email program.

- **Automation Server.** AlphaLAN++ 8.0 offers automation services allowing you to control your host application session through a program written in Visual Basic, C++, etc. A variety of properties & methods are exposed for you to set connection parameters, send startup commands to the host, send data to the host, wait for certain response from the host, get screen data, etc.

- **New User Interface for User-Defined Keys.** Version 8.0 supports a "Cool bar" that can show 4 "bands", each band corresponding to each "Level" of User Defined Keys. These keys appear as Toolbar buttons. You can display the "Label" as text or hide the labels. This presents a new look to the User Defined Key buttons. Address bar appears as the fifth band in this Cool bar. The bands can be joined together for a single row display or separated into different rows.

- **Exiting or Closing Sessions.** Version 8.0 offers new features to "close" the session or "exit" the session (close the session & exit the application). Host commands "DOS/D" and "DOS" can be used to perform these functions, respectively. You can also automatically close the session and exit the application, when you disconnect your Telnet session (or Modem Connection) to the host. You can also elect to minimize your main application window by using DOS/M command.

- **Full Screen Display.** Version 8.0 provides smoother transition to Full Screen mode and back without any flash.

- **Improved user interface for function keys and Special keys.** These keys now have new bitmaps, tool tips, and show "hot images" when highlighted.

- **Extended Keys for Keyboard re-mapper.** The ten extended keys (Insert, Delete, Home, End, Page Up, Page Down, Up Arrow, Left Arrow, Down Arrow and Right Arrow) can now be re-mapped. The Keyboard Remapper will show these keys with a prefix of "Ext".

- **UA-ftp Changes:** Provides a new "List" view in addition to the "detail" view. It displays "icons" for the popular file types. You can also specify command line arguments to go to different hosts or to use different user login information.

- **Improved Local Menus.** Right click on tool bar, cool bar provide local menus now. Right click on terminal session screen provides a new expanded local menu.

- **Sprint32 is no longer supported.** Sprint32 program for "Print Service (Printing from PC programs to your host's printer)" on RS-232 connections is no longer supported in this version.

3.0 Installation instructions

3.1 Before Installing AlphaLAN++

AlphaLAN++ 8.0 version offers a variety of new functions over previous versions. It is fully compatible to previous versions 7.XX.

If you are updating from Alpha LAN version 6.XX, please note that AlphaLAN++ 8.0 (just like the previous 7.XX versions) does not implement certain AlphaLAN 6.X features. Other features, while present in both versions, may exhibit subtle differences in AlphaLAN++ due to the nature of the 32-bit environment or for other reasons. You should test your application with AlphaLAN++ 8.0 to check for any differences in operation.

You can upgrade to AlphaLAN++ in any case, however, because AlphaLAN++ can coexist with AlphaLAN 6.2 or prior 6.X versions. Install AlphaLAN++ on the Alpha Micro host and then, on each PC workstation, install AlphaLAN++ or AlphaLAN 6.X, or even both. You can *mix and match* both generations of PC side software at the same AlphaLAN++ site. After installing AlphaLAN++, you should continue using AlphaLAN 6.2 PC-side software at workstations where any of the following apply:

1. If you are installing AlphaLAN on a DOS or Windows 3.1 PC.
2. If you want to use AlphaNET protocol for your network connections.
3. If you need to use imaging functions of inSight/AM.
4. If you need to use CoStar or the DDE interface.
5. If you need to use the Disk Server function from DOS.
6. If you need to connect Macintosh® workstations*.

* Note that Apple's Open Transport product is not compatible with any version of AlphaLAN. All new Macs use Open Transport, and not the older MACTCP Protocol on which AlphaLAN for the Mac was based. We therefore recommend that you do *not* install AlphaLAN on any new Macintosh installation requiring TCP/IP support.

3.1.1 If you are upgrading from version 7.0

The host side software of AlphaLAN++ remains unchanged from the version 7.0. Therefore, you need to install the PC side software, only. **Please note that AlphaLAN++ 8.0, PC side software, will ask you to enter a Serial Number and an activation key**. You will receive these when you contact us to obtain your 8.0 upgrade PIC.

3.1.2 If you are upgrading from a version prior to 7.0

Please install the host side software of AlphaLAN++ first, before installing AlphaLAN++ software on your PCs. The new AlphaLAN++ host Side software will work with both AlphaLAN++ PC side software as well as the older versions (6.X versions) of AlphaLAN PC side software. If you are using your PC to connect to several different host systems, please remember to update host side software to AlphaLAN++ on all these host systems prior to installing AlphaLAN++ on your PC.

3.1.3 Multiple Concurrent Sessions

AlphaLAN++ is an MDI (Multiple Document Interface) application and allows you to start multiple host sessions from within a single process. Go to "Session" menu in AlphaLAN++ window and select "New" to start another session. There is no need to start another process.

It is also possible for you to set up multiple AlphaLAN++ shortcuts (icons) or use the

same icon multiple times to do one of the following.

1. Start multiple AlphaLAN++ processes.
2. Start another session within the same process.
3. Disable multiple sessions/processes.

If you setup multiple shortcuts, you can use different configuration files for each shortcut. This allows you to specify different hosts or different startup commands for each shortcut. Please refer to the section “Starting multiple processes or multiple sessions” under “Description of new features” for more details.

As before, all TELNET sessions from a single PC will be considered as one single node of AlphaLAN++. However, the actual number of sessions may be restricted by the license for the host operating system (AMOS or Unix) and /or by the TCP/IP software license on your host (AlphaTCP license or TCP/IP license for your Unix host).

3.1.4 Startup Commands for the host

If you want AlphaLAN++ to automatically start a host program, you may enter the host commands in the "startup" tab under the "Terminal" menu in the "Settings" menu. Please note that you may start up different host applications in different sessions by entering different "start up" commands and by using different configuration files.

3.2 If you are upgrading an existing installation

Please go through the above section called “Before Installing AlphaLAN++“ prior to installing your upgrade.

3.2.1 Installing the AlphaLAN++ 8.0 update on Alpha

The host side software of AlphaLAN++ remains unchanged from the version 7.XX except for a single file, called DOS.LIT. You can simply copy this updated file to DSK0:[1,4] account. There is no need to copy any other file, or re-PIC the software.

You need to carry out the following steps only if you are updating from a version prior to 7.0. Save your current installed files on the Alpha system. If your current update has been supplied to you on a VCR or a Streamer tape, you can go to the section **"Installing Alpha side software from VCR / Streamer tape"**. If your update is supplied on a PC diskette, or if you have downloaded these files from our web site www.uasys.com, then you need to transfer the Alpha side files from your PC to your Alpha system using the previous version of AlphaLAN.

Installing Alpha side software from the PC diskette

Do not update the PC side software first as it may prevent you from transferring the Alpha side files. Insert the "Alpha" side software disk in drive A. Make sure that you are on a PC that is connected to the Alpha system using AlphaLAN software **over a serial or AlphaNET connection**. Go to AMOS prompt. Create a new temporary account on Alpha. If you use an existing account, make sure that there are no existing files in that

account. Now, enter the following at the AMOS prompt.

CALF A:\ALPHA*.* <CR>

If you downloaded the Alpha Side software from our web site, unzip the files to an empty directory (say c:\Alpha). In this case, you would change the above command to reflect the correct directory (e.g. CALF C:ALPHA*.*) where the Alpha Side files are residing. This command will transfer all Alpha side files to the Alpha account you are logged in. After all files are successfully transferred, log to OPR: account. From this account, invoke ALUPD command file to copy various AlphaLAN++ files from the temporary account to the respective accounts on “DSK0:”. The syntax for invoking ALUPD command is:

DO devn:[p,pn]ALUPD devn:[p,pn] <CR>

For example, if your temporary account (having AlphaLAN++ update files) is DSK1:[25,10], you should give the following command:

DO DSK1:[25,10]ALUPD DSK1:[25,10] <CR>

Next, give the command "VERIFY ALFLAN" from the OPR: account. This will verify that your AlphaLAN++ update has been copied correctly. Please go to the section named "PIC encoding your new software".

Loading Alpha side software from VCR / Streamer tape

If your software has been supplied to you on VCR / Streamer tape, you will need to restore all the files from it to the Alpha disk by giving the following command from OPR: account:

```
VCRRES/T DSK0:[]=ALL:[] <CR>          ; for VCR
MTURES DSK0:[]=ALL:[] <CR>            ; for Streamer
```

Next, give the command "VERIFY ALFLAN" from OPR: account. This will verify that your AlphaLAN++ update has been copied correctly.

PIC encoding your new software

Please run the IPLAN program from OPR: account. **You will need a new PIC to install the new software, if you are updating from a version prior to 7.0. If you are updating from AlphaLAN 7.x, there is no need to install any new software on your Alpha Micro host.** Otherwise, please encode your Alpha side software using the new PIC. Please contact U.A. Systems, Inc to obtain the PIC. If you are upgrading from an earlier version of AlphaLAN, your PIC may have been written below.

SSD:
ID:
PIC for IPLAN (for ___ PCs):

3.3 Installing AlphaLAN++ for the first time

If your software has been supplied to you on a VCR/Streamer tape, you will need to restore all the files from it to the Alpha disk by giving the following command from OPR: account:

```
VCRRES/T DSK0:[]=ALL:[] <CR>          ; for VCR
MTURES DSK0:[]=ALL:[] <CR>            ; for Streamer
```

Next, give the command "VERIFY ALFLAN" from OPR: account. This will verify that your AlphaLAN++ update has been copied correctly.

After this, you have to install the cable to connect your PC to your host (if required), PIC encode the Alpha Side software using the IPLAN program and modify the system INI file. For details about PIC encoding, please refer to the previous section of these release notes. Instructions for modifying your SYSTEM INI file are given as follows.

3.3.1 Modifying your SYSTEM INI file on Alpha

Please make a copy of your INI file as TEST.INI, and then make the following changes to the TEST.INI file.

1. Define a terminal driver for your PC job (Serial & Modem connections, only). Please use the correct terminal driver matching with the "Terminal Type" that you selected in AlphaLAN++ PC side software. PC.TDV terminal driver works best with the "native" emulation of AlphaLAN++. Use PCI.TDV, if you need to use *in*Sight/AM/32.
2. Include the NET and ALP drivers in a DEVTBL statement. Load these drivers in the system memory.
3. Please add the following SYSTEM statements to your INI file.

   A. SYSTEM NETBUF.LIT/N=n (replace n by the actual number of PCs using AlphaLAN++)
   B. SYSTEM PLANET.LIT (Only if you have a serial connection & plan to use Sprint32)
   C. SYSTEM TRMNAM.LIT (Optional. Refer to AlphaLAN++ user guide to see if you wish to use this function)

Please refer to AlphaLAN++ user guide for details about "defining your PC printer as AMOS Spooler" by using the ALP device.

3.3.2 AlphaTCP setup hints for using AlphaLAN++ over TCP/IP

You will need to start AlphaTCP's telnetd program to get TELNET connections from AlphaLAN++. If you are using AlphaTCP 1.5, please add a –n switch to the telned line in the config file. For all AlphaTCP versions, you will need to start ftpd program (with –n switch for Unix style directories) to do file transfers over TCP/IP. You will need to start LPR on your Alpha Micro host and perform the related set up as given in AlphaTCP Administrator's Guide, if you wish to print to the PC printer(s) by using the optional WinLPD program. Similarly, you will need to start LPD on your Alpha Micro host, if you wish to print from PC applications to AMOS printers over TCP/IP.

3.4 Installation of AlphaLAN++ software on your PC

AlphaLAN++ software for the PC comes to you in one of the following 3 ways.

1. On 3 ½" disks.
2. On a CD-ROM.
3. Electronically through a self-extracting EXE by email.

Insert the floppy in Drive A. Go to "Start" button, select "Run", type "A:SETUP" and hit the "Enter" key, if you are installing AlphaLAN++ from the floppy drive. For CDROM installation, follow the above procedure but run the "SETUP" program from the CD-ROM. For installing AlphaLAN++ from the self-extracting EXE, please run that EXE by typing the name of the EXE (generally ALAN80.EXE). Follow the screen prompts to finish the installation.

Important Note for Win 95 Users: If you are installing AlphaLAN++ on a Windows 95 system, you will need to update HTML help. **Please remember to say "No"**, when the installation program asks you if it should **restart your system**. In the end, the installation program will ask you if you wish to update HTML Help. Say "Yes". After HTML help has been installed, you should re-start your computer by going to the "Start" menu.

Serial Number & the Activation Key: Starting from version 8.0, you will need to enter a serial number and the activation key to install your PC side software. You can obtain this information from us. This information looks as follows.

Serial Number: NNNNNNN (N is a number between 0 and 9, total digits between 4 and 7)
Activation Key: AAAAA – AAAAAAAA – AAAAA (A is an alphanumeric character (0-9, A-Z), Total digits: 5-8-5)

4.0 Description of New Features

AlphaLAN++ introduces many new features that add power and flexibility to PC connectivity in the Unix and AMOS environment. For details of these features see the AlphaLAN++ documentation.

These release notes describe the salient new features over the previous version of AlphaLAN++.

4.1 Host Programming Interface for various functions

AlphaLAN++ offers an interface for host programs to use a variety of functions. **These functions work in all emulations** and are described below. All functions start with a standard "lead-in" sequence, **ESC ESC ~**. This should be followed by the specified sequence (as given within brackets against each function, followed by parameters, if any. The parameters may have to end by sending a "Delimiter" character, if so specified in the command string during the function definition. The standard delimiter is the character DEL (0x7F in hex or 127 in decimal) OR US (0x1F in hex or 31 in decimal). We will use ***del*** to refer to this delimiter character.

4.1.1 Start a browser session with the specified URL (+5 *urlparam del*)

This function allows you to start a Browser Session. *urlparam* is an optional string parameter that defines the URL address for the browser. This command must end with a *del* character.

4.1.2 Execute a command specified by a button in the specified UDK file (+6 *del udkfilename del keyval del del*)

This function allows you to execute previously defined functions using “User Defined Keys” function of the application. *udkfilename* refers to the name of the UDK file (say user.udk or “C:\Program Files\UASystems\AlphaLAN++\user.udk”). *keyval* refers to the key number(decimal 48(0x30 hex) plus a decimal number 0-63 corresponding to 16 user keys for each of the four Levels). Please use the delimiter as specified above in the command string.

4.1.3 Write host data to a PC file (+7 *del pcfilename del data del a del*)

This function allows you to write data to a PC file from your host application. *pcfilename* refers to the name of the PC file. *data* refers to the data that you want to write to this PC file. **The maximum length of the data is 10 KB**. Please notice that you can specify an **optional switch** (letter a or A), if you wish to append this data to the file. Otherwise, the existing data in the file, if any, would be overwritten.

4.1.4 Read data from a PC file (+8 *pcfilename del*)

This function allows you to read data from a PC file. *pcfilename* refers to the name of the PC file. The contents of the file would be sent to the host application.

4.1.5 Copy data to PC clipboard (+9 *datastring del*)

This function allows you to copy a datastring to the clipboard. *datastring* refers to the data string that you want to copy to the clipboard. The data string can be a maximum of 256 characters.

4.1.6 Read (Paste) data from PC clipboard (+B)

This function allows you to read data from the clipboard. You can read the entire data from the clipboard. This read operation may be cancelled by pressing Control C.

4.1.7 DDE Interface(See below)

This interface allows you to pass information back & forth (called conversation) between your host application (the terminal emulator acts as **DDE Client**) and a Windows Applications (**DDE Server**) like MSWord or Excel.

4.1.7.1 Defining DDE Terminology and Methods

Application Name (or Service Name): The DDE Server (see above) that participates in the conversation is identified by its *Application name* (or *Service name*). This is usually, but not always, the name of the application’s executable file, without the .EXE extension. Actually, the *Application name* is the name by which Windows Task Manager recognizes the application. For example this *Application name* is WINWORD for MSWord and EXCEL for MS Excel.

Topic Name: When you start the conversion with a DDE Server, you also need to specify a *Topic name*. In the majority of cases, the data of interest is contained in a document (file) that you specify as the DDE Topic name. All DDE Server applications are required to have a Topic name *System*. Besides using the standard *System* Topic, you can use the name of the file as the topic. e.g. "c:\excel\worksheets\sheet.xls" or

"c:\My Documents\testdde.doc".

Item Name: The item name is the method by which the server application specifies a particular piece or set of data, the data item. For excel, you can use "**R2C3**" (2nd Row, 3rd Column) or "**R1C1:R3C3**" (specifies 9 cells from the first 3 rows and three columns of the Excel spreadsheet) as item names. For MSWord, items are **Bookmarks** denoting the beginning and the end of the text. The bookmark can also specify a Table in the MSWord document.

DDE Methods (or commands): After initiating a conversation with a DDE Server (**DDEInitiate**), you can update data in MSWORD Tables, Excel worksheets, etc. from your host program using **DDEPoke** (sends unsolicited data to the DDE Server) method. You specify the item name and its value when using the DDEPoke command. You can also get the data from these tables to your host programs by using **DDERequest** (gets the current value of the requested item). The value is sent back to your host program as if some one typed it from the keyboard. You can also start a "hot" data link to an item in the DDE Server by sending the **DDEAdvise** command. This allows you to create a “hot” (continuous) data link so that your host application receives the data whenever some one changes the spreadsheet/Word tables. You terminate the “hot” data link by sending the **DDEUnadvise** command. You can also execute commands in the Server application by sending the **DDEExecute** command. The commands are often the menu choices or keywords in the application’s macro language. Most DDE servers that recognize execute commands require that you enclose the commands within square brackets. For example, the following command creates a new Excel Chart from the data contained in columns 1 through12 of row 2 of the worksheet.

“[Select(“ & Chr$(34) & “R1C1:R2C12” & Chr$(34) & “)][New(2,2)]”

Note*: Excel requires that selection coordinates or range names be enclosed within single quotation marks (chr$(34)).*

Finally, you can end the DDE conversation by sending the **DDETerminate** command.

There is a standard item called "SysItems". If you send DDEREQUEST, with "*System*" topic, and "*SysItems*" as the item, then you will receive a list(TAB separated) of data items that application supports. Most application support items *Topics*, *Status*, *and Format* besides *SysItems*. DDEREQUEST on *Topics* as item will give you the list (TAB separated) of all open document files, plus *system*.

4.1.7.2 DDE Interface commands

The Escape codes for various DDE commands are given as follows.

1. DDE**I**nitialize(**+A** or **+I** *del AppName del TopicName del str1 del … strn del del*).

   AppName is the application name of the DDE Server running on the PC. TopicName is the Topic for the DDE conversation. Str1 del …. Strn del are the execute command strings (followed by the delimiter del) and are optional. Please note that you can send them later by using another Escape code as specified next.

   After the DDE Conversation has been initiated, if the Server sends back any

DDEData, it would be sent back to the host as if this data was entered from the PC Keyboard.

2. DDE**E**xecute (**+E** del *str1 del … strn del del)*

   Str1 del …. Strn del are the execute command strings (followed by the delimiter del). Please note that you must send the DDEInitialize command first before sending this command. These execute string can also be included within the Initialize command. If the Server sends back any DDEData, it would be sent back to the host as if this data was entered from the PC Keyboard.

3. DDE**P**oke (**+P** del strItem del strValue del)

   After initiating a conversation with a DDE Server, you can update data in MSWORD Tables, Excel worksheets, etc. from your host program using DDEPOKE (sends unsolicited data to the DDE Server). You specify the item name and its value when using the DDEPOKE command.

   The client renders the item to be sent and sends the WM_DDE_POKE message. StrItem refers to the item's name and strValue is its value that you want to send to the DDE Server.

4. DDE**R**equest (**+R** del strItem del)

   To retrieve an item from the server, the client sends the server a DDERequest message specifying the item. StrItem refers to this item's name. The value (a string), once received from the DDE Server, would be sent back to the host as if it was entered at the PC Keyboard.

5. DDEAd**v**ise (**+V** del strItem del)

   A client application can use DDE to establish a permanent link to an item in a server application. After such a link is established, the server sends periodic updates of the linked item to the client, typically, whenever the value of the item changes. Thus, a permanent data stream is established between the two applications; this data stream remains in place until it is explicitly disconnected.

   The client initiates a data link by posting a DDEAdvise message to the DDE Server specifying the item name. StrItem refers to this item's name. The value (a string), once received from the DDE Server, would be sent back to the host as if it was entered at the PC Keyboard

6. DDE**U**nadvise (**+U** del strItem del)

   The client terminates the specific (permanent) data link by posting a DDEUnadvise message to the DDE Server specifying the item name. StrItem refers to this item's name. You must call DDEUnadvise for every item that you sent DDEAdvise message for. Please note that this command terminates a specific data link. It does NOT terminate the DDE conversation.

7. DDE**T**erminate (**+T**)

The client can terminate the DDE Conversation to the server by issuing this command. You must issue this command to terminate the DDE conversation when you are done with the DDE conversation.

4.1.7.3 Sample code for the integration of AMOS (or Unix) and PC applications:

Example 1: Send Customer's (or a patient's) information from your host application to MSWord document (wordpoke.doc and wrdpok.bas files)

We will use the DDEPoke command to achieve this. The sample file, WordPoke.doc, is a sample letter that is sent to a patient reminding them about the appointment at the doctor's office. This letter has three bookmarks as follows.

Patient – Patient' name.
Address – Patient's address

The following code illustrates how you can send the information for the above 2 bookmarks to this word file.

```
!**** source code as per wrdpok.bas file **********

map1 delimiter,f
delimiter=126

!Optional: Start the MSWord Application & open the customer's document file
!Please correct paths for the MSWORD program file and the sample file.
?chr(27);CHR(27);"~%d";"d:\Program Files\Microsoft Office\Office\Winword.exe "
;chr(34);"c:\Program Files\UASystems\AlphaLAN++\WordPoke.doc";chr(34);chr(13);
chr(127);chr(27);chr(27);"~m";

! The above command forces a dialog box "Press return to continue". Wait
! for MS Word to start and load the document before pressing return.

!send DDE Initiate
?chr(27);CHR(27);"~+A";
?chr(delimiter);
?"WINWORD";
?chr(delimiter);
?"c:\Program Files\UASystems\AlphaLAN++\WordPoke.doc";
?chr(delimiter);
?chr(delimiter);

!Send DDE Poke Messages
!First - Address
?chr(27);CHR(27);"~+P";
?chr(delimiter);
!item name
?"Address";     ! Name of the book mark in the MSWord file.
?chr(delimiter);
!Item value
?"Jane Doe";chr(13);"1234 Silverado Street";chr(13);"MyTown, CA 12345-6789";
?chr(delimiter);
```

```
!Send DDE Poke Message
!Second - Name
?chr(27);CHR(27);"~+P";
?chr(delimiter);
!item name
?"Name";        ! Name of the book mark in the MSWord file.
?chr(delimiter);
!Item value
?"Jane Doe";
?chr(delimiter);

!Send DDE Terminate
?chr(27);CHR(27);"~+T";
```

Example 2: Update MS Excel spreadsheet data from a host application (ExcelPoke.XLS and exlpok.bas files).

We will again use the DDEPoke command to achieve this. The sample file, ExcelPoke.xls, is a sample spreadsheet. We will change the value in cell (B11) from 10000 to 15000. The following code (exlpok.bas) illustrates this.

```
!**** source code as per exlpok.bas file **********

map1 delimiter,f
delimiter=126

!Optional: Start the MSExcel Application & open the customer's file
!Please correct paths for the MSExcel program file and the sample file.

?chr(27);CHR(27);"~%d";"d:\Program Files\Microsoft Office\Office\Execl.exe "
;chr(34);"c:\Program Files\UASystems\AlphaLAN++\ExcelPoke.xls";chr(34);chr(13);
chr(127);chr(27);chr(27);"~m";

! The above command forces a dialog box "Press return to continue". Wait
! for MS Excel to start and load the sheet before pressing return.

!send DDE Initiate
?chr(27);CHR(27);"~+A";
?chr(delimiter);
?"Excel";
?chr(delimiter);
?"c:\Program Files\UASystems\AlphaLAN++\ExcelPoke.xls";
?chr(delimiter);
?chr(delimiter);

!Send DDE Poke Messages
!cell Row 11, Column B, Advertising, Q1
?chr(27);CHR(27);"~+P";
?chr(delimiter);
!item name
?"R11C2";    ! the cell or the cell range
```

```
?chr(delimiter);
!Item value
?"15000"; ! change from 10000 to 15000
?chr(delimiter);

!Send DDE Terminate
?chr(27);CHR(27);"~+T";
```

Example 3: : Send Customer's (or a patient's) information from your host application to any MSWord document (any opened Word Document and wrdexe.bas files)

This example uses DDEExecute command to achieve this. You can open any MSWord file. The following code (wrdexe.bas) would insert Customer name & address at the cursor position.

```
!send DDE Initiate                                          0
?chr(27);CHR(27);"~+A";                                       0
?chr(delimiter);
?"WINWORD";
?chr(delimiter);
?"System";      ! Using "System" topic. This will work with all Word files
?chr(delimiter);
?chr(delimiter);

!Send DDE Execute strings - 2 strings, name & company
?chr(27);CHR(27);"~+E";
?chr(delimiter);
?"[Insert " + CHR(34);
?"Jane Doe";
?CHR(34) + " + CHR$(13)]";
?chr(delimiter);
?"[Insert " + CHR(34);
?"Doe's Company";
?CHR(34) + " + CHR$(13)]";          ! You can combine & put it up as a single string.
?chr(delimiter);
?chr(delimiter);

!Send DDE Terminate
?chr(27);CHR(27);"~+T";
```

Other Programming Tips:

You can use the DDERequest command to receive data from Word documents or Excel spreadsheets in your host applications. You can even set up a "hot" link to these Word files or Excel spreadsheets to receive "live" data continuously whenever there is a change by using DDEAdvise command (live stock quotes, ???). There are many interesting host and PC application integration possibilities here.

4.1.8 Launch Windows applications *(% cmd pcprogramname1 cr … pcprogramnamen cr del ESC ESC~m*)

You may launch "Windows application(s)" from the host by sending the above command string. *cmd* refers to a 'command' letter as defined below.

d - Puts up a "Press any Key to continue" Dialog box.
/ (slash) - Puts up the above dialog box only when more than one program is being started.

Pcprogramname1 refers to the actual command string for the program being executed. After the program name, you must have cr (0x0D in hex), then the delimiter del, followed by ESC ESC ~ m. **Do not put any spaces in the above sequence** except when specifying arguments as a part of the program name. Please make sure that the Windows application's directory is included in the path statement in the autoexec.bat file. Otherwise, include the complete path in the program name.

4.1.9 Close the session *(d*)

This would cause the session to close. It is same as closing the session from the "close" button of the ChildFrame. The application will not exit.

4.1.10 Close the session and exit the application *(m*)

This would cause the session to close and the application to exit, if you are exiting the last session and no other sessions are active.

4.1.11 FTP file transfer (t dir del hostdir del srcfilename del destfilename del options del del)

This function allows you to transfer files between your host computer and your PC by using ftp protocol. You should be using the TELNET connection over TCPIP in order to use this function. Please do the following set up.

In the Settings, Terminal, Connection, Native File Transfer, enter a valid ftp user name and password.

In the above command string, ***dir*** is defined as follows.

Use **l** (or **L**) for local transfer (Copy to Local PC from host) .
Use **r** (or **R**) for remote transfer (Copy to Remote host from PC)

Hostdir refers to the host directory (e.g. /usr/bin for unix hosts and DSK0:[25,10] or SYS: for AMOS hosts)
Srcfilename refers to the name of the source file. If you are specifying a PC file name (for copying to your host), you may include the full path name. Wild carding is allowed to transfer multiple files.

Destfilename is optional and may be specified when transferring single files.

Options are optional. The following options are supported.
/O – Overwrite if file exists
/N – Skip (No-delete) if file exists
/Q – Confirm (Querry) before transfer
/D – Delete Source file after transfer is complete.

/P – Print the source file to PC Printer (only if the direction is **local**)
/S – Silently transfer files. Progress bar & error messages are suppressed.

for example

ESC ESC "~tl" 0x7f "/user/john" 0x7f "myfile.txt" 0x7f "" 0x7f "" 0x7f 0x7f
This will transfer myfile.txt from /usr/john directory on the Unix host to the current working directory on the PC.

4.1.12 Save Rectangular Region/Box *(v line col*)

This command allows you to save a rectangular region by using the “Save/Restore area” function of the emulator. Please note that you can do MULTIPLE save/restores and you can increase the size of this stack by increasing the number of Save/Restore Pages (default is 4) in the emulator’s settings (Settings, Screen).

You need to position the cursor to the top left corner of this region and then issue the above command. Here line refers to the number of lines (length) and col refers to the number of columns (width) in the specified region. You need to add 0x20 to the column/row numbers. Therefore, column 1 or row 1 is 0x21 (ASCI !). Also columns greater than 95 need 0x1F(31 in decimal) and offset by 95.

For example, if you are on row 1 and column 1, ESC ESC ~v 8 p will save the entire 24X80 screen.

4.1.13 Restore Rectangular Region/Box *(w line col*)

This command allows you to restore previously saved rectangular region. Please refer to the previous section for more details. For example, if you are on row 1 and column 1, ESC ESC ~w 8 p will restore the previously saved (entire 24X80) screen.

4.1.14 Minimize the main application window (z)

This function allows you to minimize the main application window.

4.1.15 Set the screen width to 132 columns (1)

This function allows you to change the screen width to 132 columns.

4.1.16 Set the screen width to 80 columns (2)

This function allows you to change the screen width to 80 columns.

4.2 FTP File Transfer - new features

UA-FTP, Version 8.0, supports several new features as given below.

4.2.1 Starting UA-FTP by using a URL

You can specify the following URL to start UA-FTP, now.

ftp://ftpusername:ftppassword@hostipaddress:ftpport

Please replace the ftpusername, ftppassword, and hostipaddress with your own information when entering the above URL. “ftpport” is optional and may be left blank. Remember to include the last “:” before the ftpport. For example

ftp://john:mypassword@uasys.com:

You can specify the above URL to start UA-FTP in the following different ways.

1. Use the **Address bar**: You can type the URL in the address bar of any AlphaLAN session (the terminal session or the browser session).

2. Set up “UA-FTP” **shortcut** specifying the URL at the end of the command as given below.

   “C:\Program Files\UASystems\AlphaLAN++\uaftp.exe” ftp://anu:mypass@192.10.9.1:

   Please note that this facility allows you to setup multiple “UA-FTP” shortcuts each with a different URL to connect to different hosts, directories, etc.

3. Start FTP URL from within your host application by using a special Escape code. Please refer to the section “Starting a URL from host application”. An Alphabasic example is as follows.

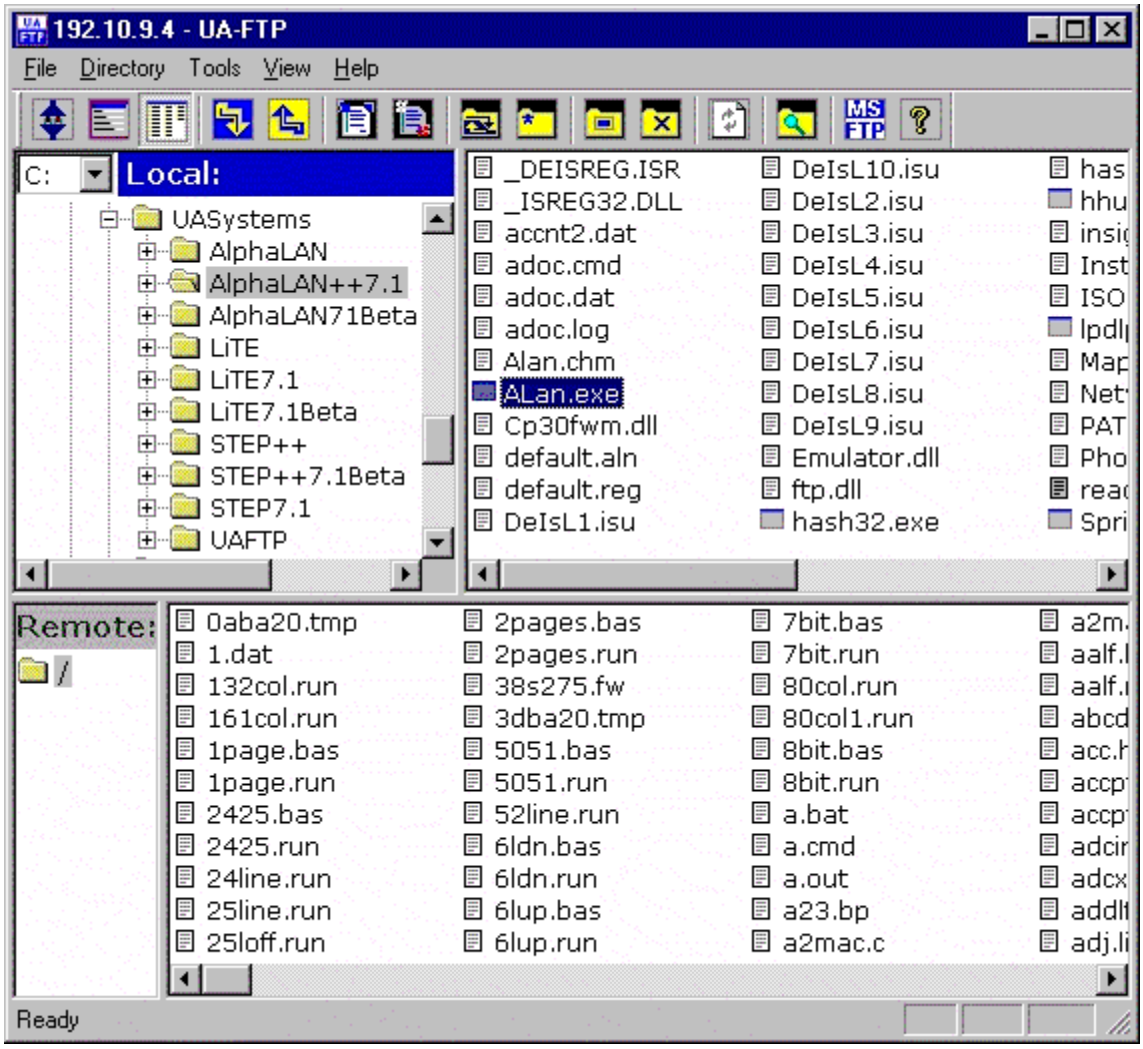
   ?chr(27);chr(27);"~+5";"ftp://anu:anu@192.10.9.4:";chr(127);

   Please remember to include the delimiter (0x7F or dec 127) at the end.

Please note that the FTP user name, password, hostipaddress and ftpport are set up on your host computer. Please contact the system administrator to get this information or refer to your host side TCP/IP software release notes for instructions about setting these details.

4.2.2 The List view

UA-FTP 8.0 offers the “list view”. This view shows the complete list of files without showing details like date, size, etc. The list view shows just the list of files along with a bitmap as shown below.



4.3 New user interface for User Defined Keys

These keys are now displayed in a “rebar” (or “cool-bar”) as four “bands” of sixteen keys each. Four bands correspond to the four “Levels”. Each band shows up to sixteen ‘user defined keys’ as “buttons” numbered 1 though 16. The band will automatically size itself based on the number of keys that you “define (program)” for that level. These bands may be moved around inside the rebar by using the ‘gripper bar’ shown at the beginning of each band. You can also show them in different rows. However the band order is fixed. In other words, band 2 would always appear after band1 and so on. Each band’s name corresponds to the “level name” that you defined. If undefined, these bands have the names as level1, level2, and so on.

UDK buttons show the button number (image) on the top and display the “label text” below the image. If you check “Hide Image”, the UDK buttons would be shown with the label text but the top part (image) showing the button numbers would be hidden and the height of the buttons would reduce. When the button number (image) is being shown, you can show or hide the “label text”. The label text will always be displayed as a tool tip when you move the mouse over the button.

To show UDK bands in the “Rebar”, go to Tools, Customize, User Define Keys and program the

User Defined Keys. Remember to assign a “Label” to each key that you define. Save these definitions to a “.UDK” file. Then go to the "Miscellaneous" tab of "Keyboard" options under "Settings". Check the “Show” box and then select this .UDK file.

You can show/hide various bands in the rebar by using the “UDK” menu under “View” and checking/un-checking the appropriate band. These bands are named using the “Level name” that you specify when you define the keys.

4.3.1 Programming the User Defined Keys (UDK Editor)

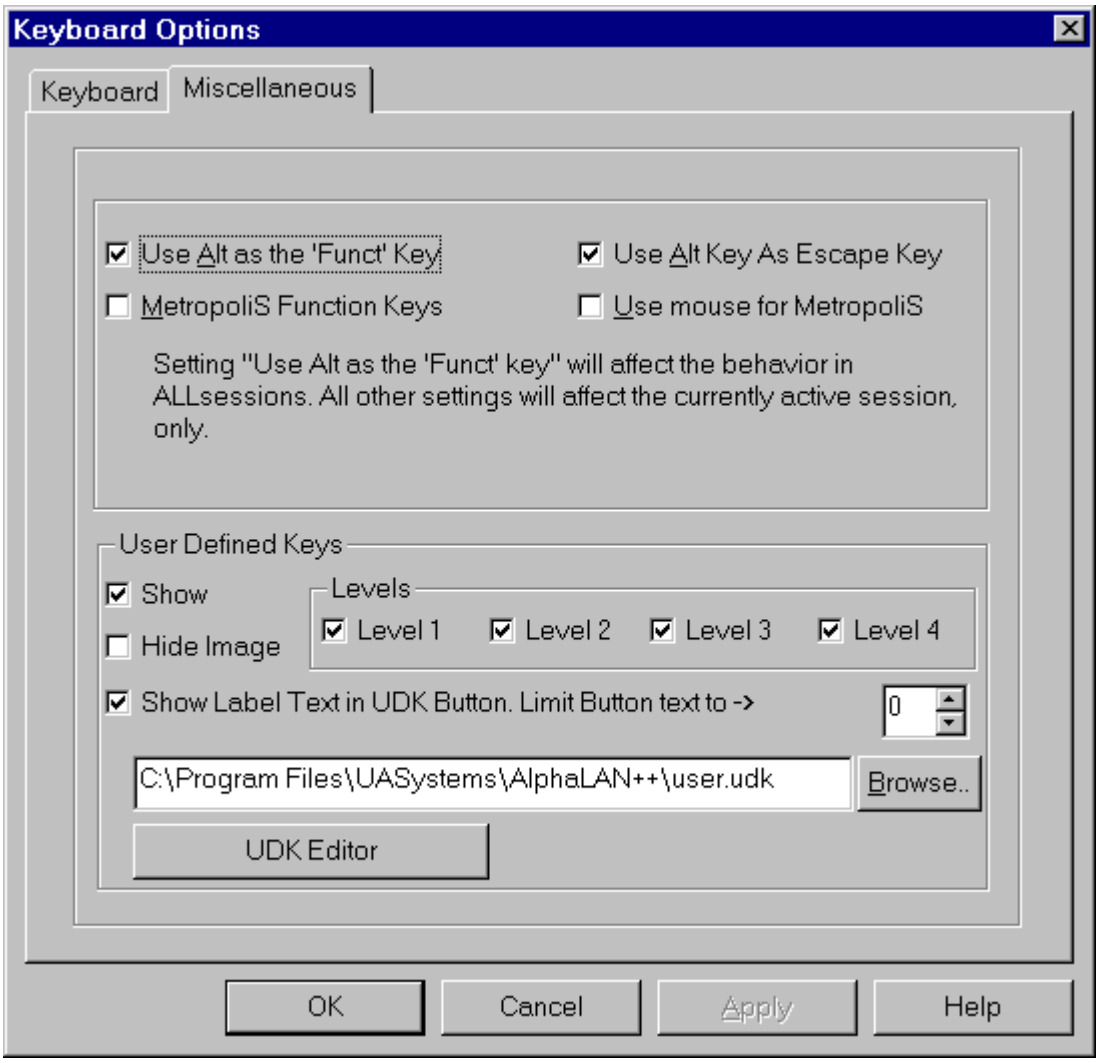
The programming of User Defined Keys remains un-changed.

4.3.2 Saving and loading of User Defined Keys

This remains un-changed.

4.3.3 Enabling the User Defined Keys

After programming the User Defined Keys and saving them into a .UDK file, you must enable them. Go to "Settings", "Keyboard" and select the following "Tab" called "Miscellaneous".



Under “User Defined Keys”, check the “Show” box and then select the .UDK file. By default, UDK buttons show the button number (image) on the top and display the “label text” below the image. If you check “Hide Image”, the UDK buttons would be shown with the label text but the top part (image) showing the button numbers would be hidden and the height of the buttons would reduce.

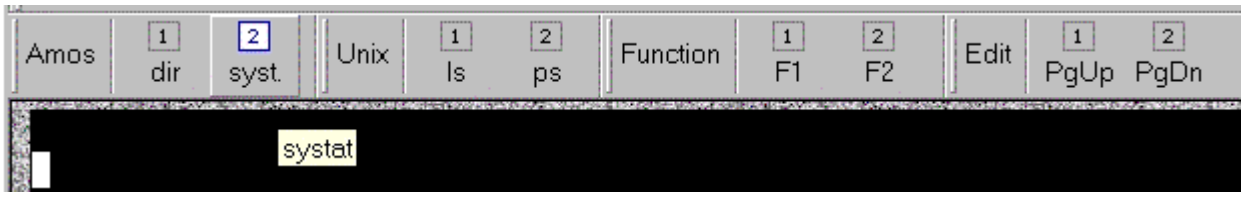
Otherwise, you can un-check the “Show Label Text in UDK Button” to reduce the height of the buttons. The UDK buttons would not show the label text in this case. This label text will only get displayed as a tool tip when you move the mouse over the button. However, it may not be as convenient as seeing the label text on all buttons at the same time.

All buttons have the same width. When the label text is being displayed, then the button width would depend on the length of the longest label text. You can limit the button width to a certain number of characters by selecting this number in the above screen. The value of “0” means that there is no limit and the width depends on the length of the longest text label.

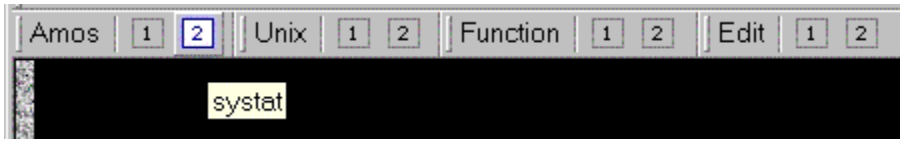
4.3.4 Using the User Defined Keys

AlphaLAN++ displays “user defined keys” as “buttons” in four bands of sixteen buttons each shown in a “rebar (or cool-bar)” along with the “address-bar”, if shown. Each band’s name corresponds to the “level name” that you defined. If undefined, these bands have the names level1, level2, and so on. You can select a key by clicking on the corresponding button.

The following rebar shows the four bands named Amos, Unix, Function and Edit. Each band has two keys (buttons). Mouse is on Amos band, second button with label text as “systat”.



If you check “Hide Image”, the buttons would get smaller. You will see just the label text (dir, syst, etc.). The square button images showing the button numbers (1,2…) would not be displayed. The buttons would also get smaller, if you uncheck the “Show Label Text in UDK Button”. You will NOT see the label text, just the images as shown below.



4.4 Extended Keys in Keyboard Remapper

The ten extended keys (Insert, Delete, Home, End, Page Up, Page Down, Up Arrow, Left Arrow, Down Arrow and Right Arrow) can now be re-mapped. The Keyboard Remapper will show these keys with a prefix of “Ext”.

You can also turn on the Caps Lock and/or Num Lock keys for the terminal emulator. Please check the relevant boxes in Settings, Keyboard tab.

4.5 Starting a URL (Browser, FTP, Mail)

You can start a "browser" session in one of the following three ways.

1. By typing the URL in the address bar of your terminal session.
2. By marking the URL text on your host screen using your mouse. Then select "Start URL" under "Edit" menu (or use the local menu by clicking the right mouse button).
3. By using the following Escape code from within your host application.

   Esc Esc ~+5 URL 0x7F

   URL refers to any valid URL. Please remember to include the delimiter (0x7F or dec 127) at the end. For example

   Esc Esc ~+5 www.uasys.com 0X7F

   The above escape code will start a browser session within AlphaLAN++ and connect you to uasys.com home page.

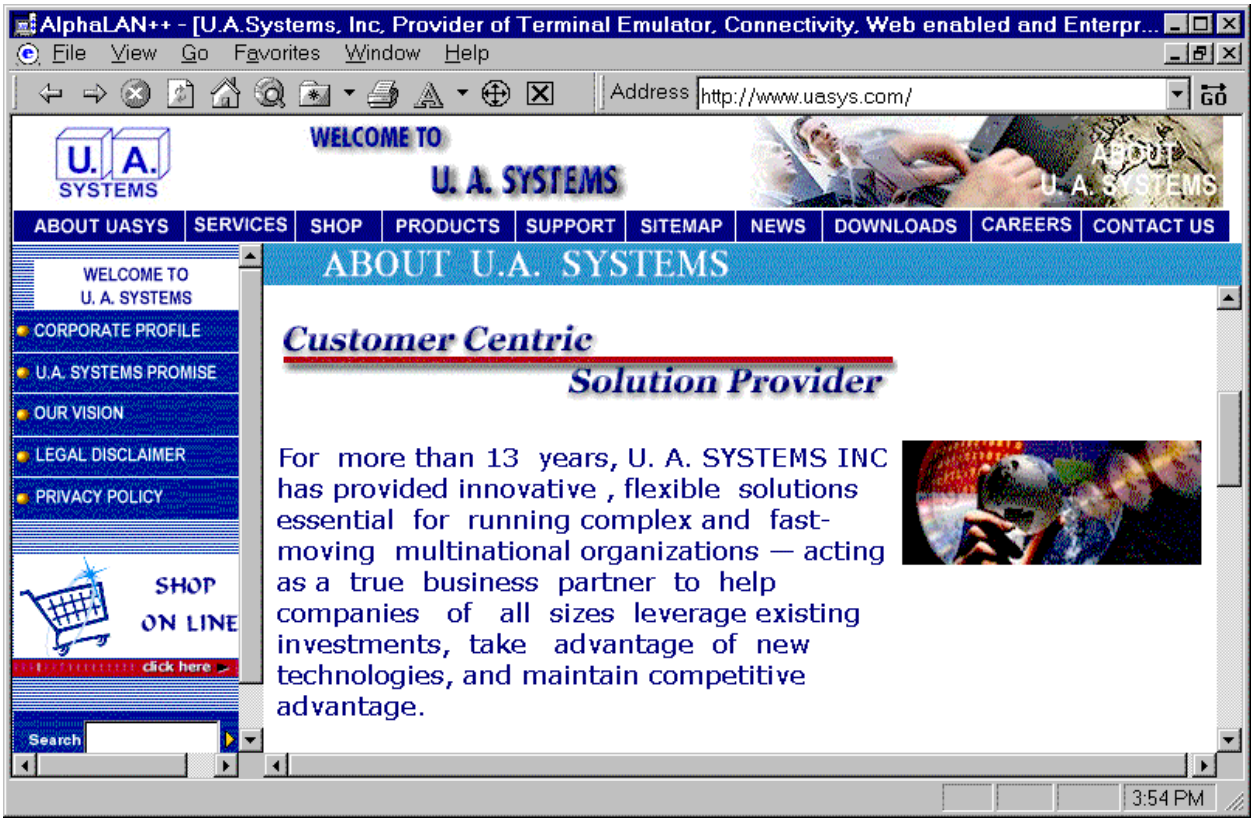
If your URL starts as **mailto://** or has an email address, AlphaLAN++ will start your default mail program with that mail address. If your URL starts as **ftp://** then AlphaLAN++ will start UA-FTP program. Please refer to the section "Starting UA-FTP by using an URL" for details.

4.6 Browser Sessions

Version 8.0 allows you to have browser sessions within AlphaLAN++. Just like the terminal session, you can have a browser session. Multiple browser sessions are allowed. You can switch between multiple browser & terminal sessions by using the Control+Tab key.

You can start a browser session by using the menu, or typing the “URL” in the Address bar of the terminal session, or by using a special “Escape Code” from your host program. **A new command line option (/b) is available to set up a shortcut to start the application with a browser session instead of a terminal session.**

The browser session looks like the following.



The browser session offers a “Navigation Toolbar” with the following browser functions.

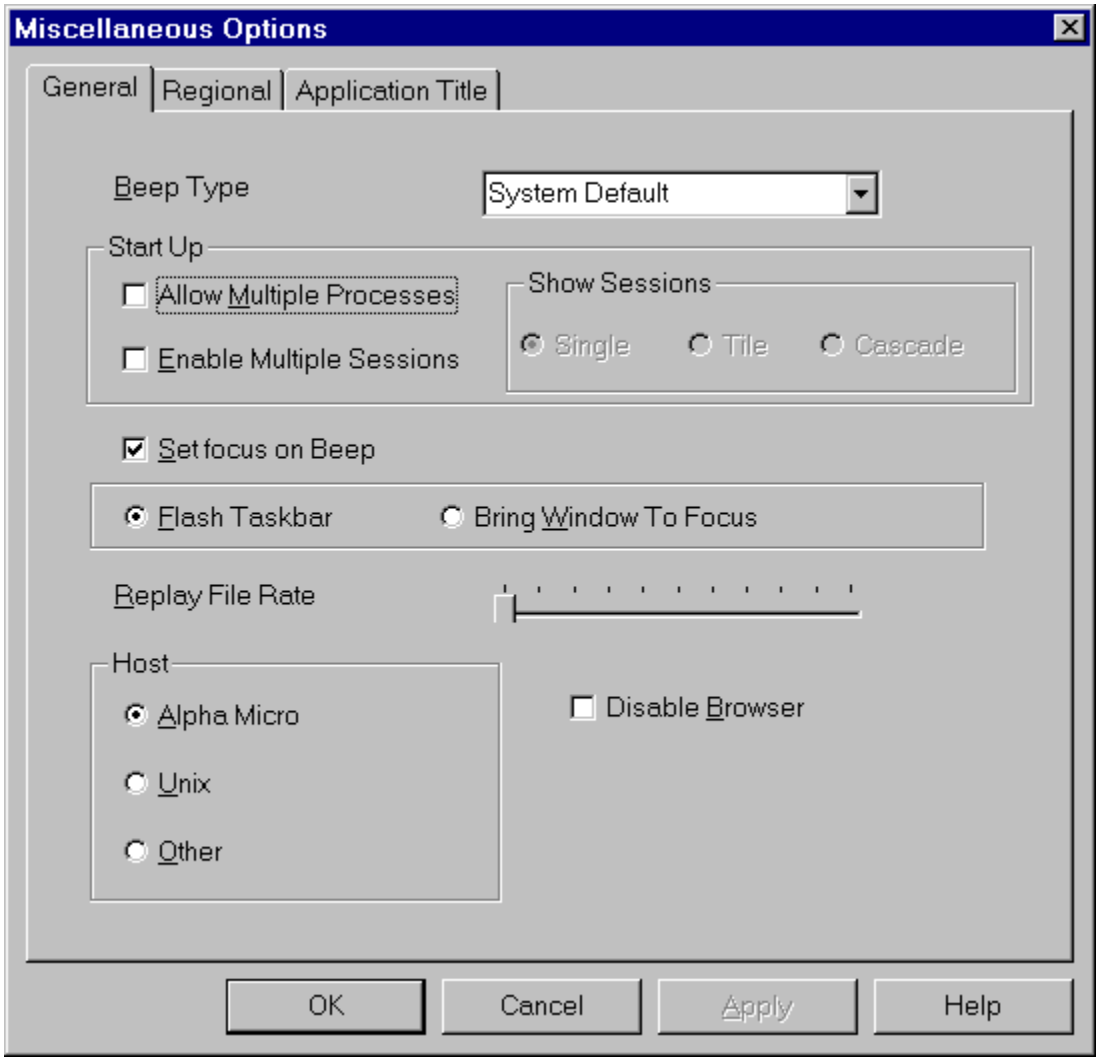
1. Back & Forth
2. Stop
3. Refresh
4. Start page
5. Search
6. Favorites
7. Print
8. Font
9. Full screen
10. Address Bar
11. Go Button

Most of the menu commands and options are self-explanatory and are similar to the ones found in Internet Explorer. "Open in new Window" option under "View" menu has the following three settings.

1. Internet Explorer: Will start a new Internet Explorer window. This is the default selection.
2. Browser Session: Will start a new Browser session in AlphaLAN++.
3. Disable: Will not start a new window. It will be shown in the current browser session.

4.6.1 Disabling the browser access

You can disable the browser access completely by checking “Disable Browser” in the “General” tab in “Miscellaneous” under “Settings”. This would prevent all access to the browser.



Hint: You can lock the “Miscellaneous” settings using a password. This would permanently disable browser access.

4.7 Automation Server

AlphaLAN++ 8.0 offers automation services allowing you to control your host application session through a program written in Visual Basic, C++, etc. Here is a list of properties & methods exposed by AlphaLAN++ automation server.

Properties:

boolean SessionStarted;

Methods:

void Activate(); void Deactivate();

Deactivate() minimizes the application. Activate() brings it into the focus.

boolean OpenFile(BSTR strFileName)

Starts the session using the specified configuration file.

void Quit();

Exits the application.

void SendData(VARIANT varData, boolean bWithoutDelay);

Sends the data to the host.

void Show(short nCmdShow);

Minimizes, maximizes the window.

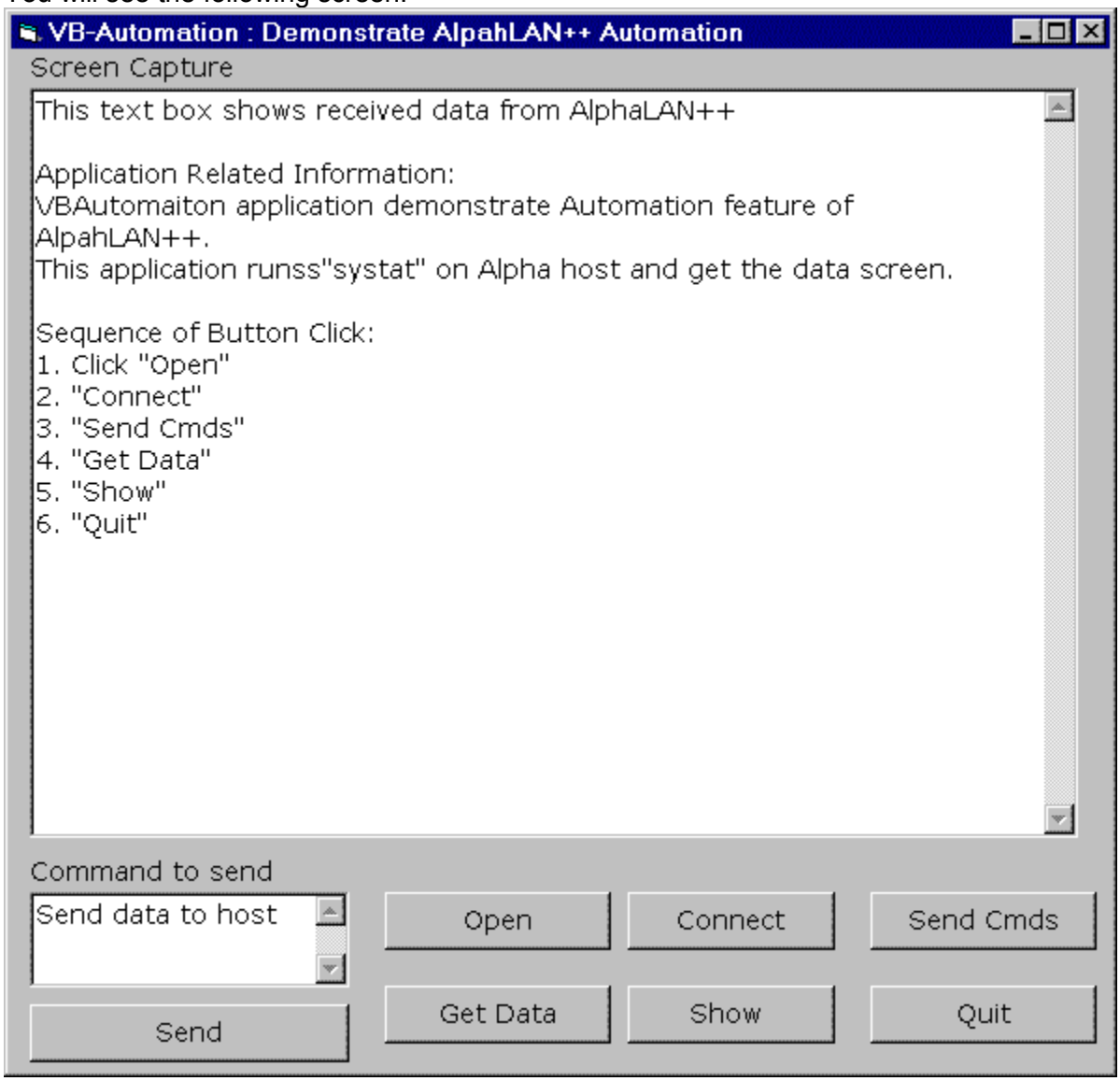
boolean WaitForSequence(BSTR strSequence, long nTimeOut);

Waits until the specified sequence is received from the host.

BSTR GetData(short startRow, short startCol, short endRow, short endCol, boolean bRectangularRegion);

Gets the data from the terminal screen.

One could write a visual basic application and interface it to the host application by using the above functions. A sample VB application, along with the source code, is included in the “automation” sub-directory of the application. To start this sample application, go to “Start”, “Programs”, “AlphaLAN++” and select “Alan VB Automation”. You will see the following screen.

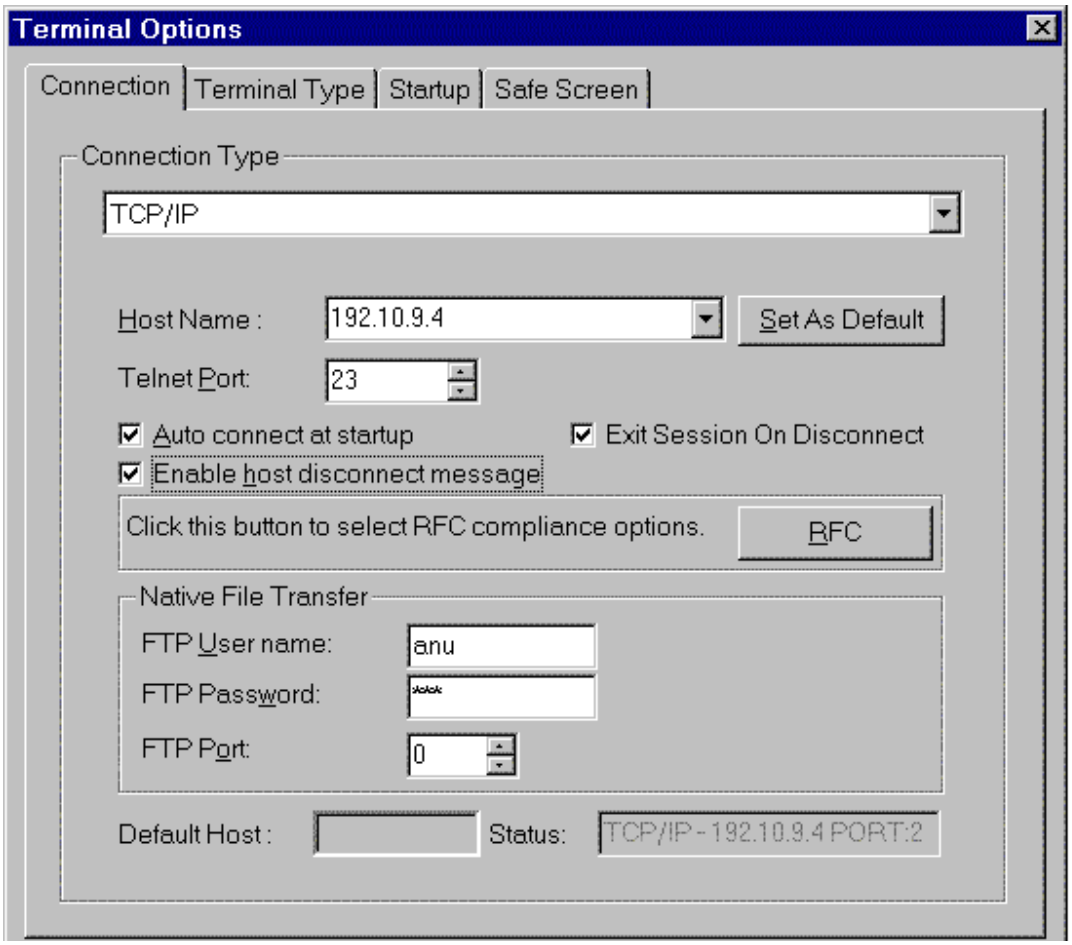


Click on “Open” to start the AlphaLAN session. Next, click on “Connect” to connect it to your AMOS host. Then click on “Send Cmds” to send a command to the host (in this case it sends the SYSTAT command to AMOS host). “Get data” will capture the entire host screen. You can also type any host command in the “Command to send” box and click on the “Send” button to send it to the host. This application demonstrates how you can control a host application from your VB program and get/ send data between your VB application and the host.

4.8 Exiting/Closing/Minimizing Sessions

AlphaLAN++ 8.0 supports the following

1. **Close a session and exit the application**: This would cause the session to close and the application to exit, if you are exiting the last session and no other sessions are active. The host command DOS (or FRCDOS) can be used to perform this action. Otherwise, you can issue the following Escape code from your host application, ESC ESC ~m.
2. **Close a session**: This would cause the session to close. It is same as closing the session from the “close” button of the Child Frame. The application will not exit. The host command DOS/D can be used to perform this action. Otherwise, you can issue the following Escape code from your host application, ESC ESC ~d.
3. **Exit session on Disconnect or Hang-up of a session**: User can select this option in the “connection” tab of “Terminal” settings. Separate options are available to you for TCPIP (Telnet) sessions and Modem sessions to automatically “Close the session and exit the application” whenever you disconnect (or hang up) the session.



4. **Host Disconnect for Telnet Sessions**: If the host disconnects a Telnet session, you will see a Message Box informing you about the disconnection. If you do not wish to see this message, please un-check “Enable host disconnect message in the above screen.
5. **Minimizing the Main Window**: If you wish to minimize the application window from your “Host applications”, you may do so by using DOS/M command from the host prompt. You will need to use the new DOS command to use this option. Otherwise, you can issue the following Escape code from your host application, **ESC ESC ~z**.

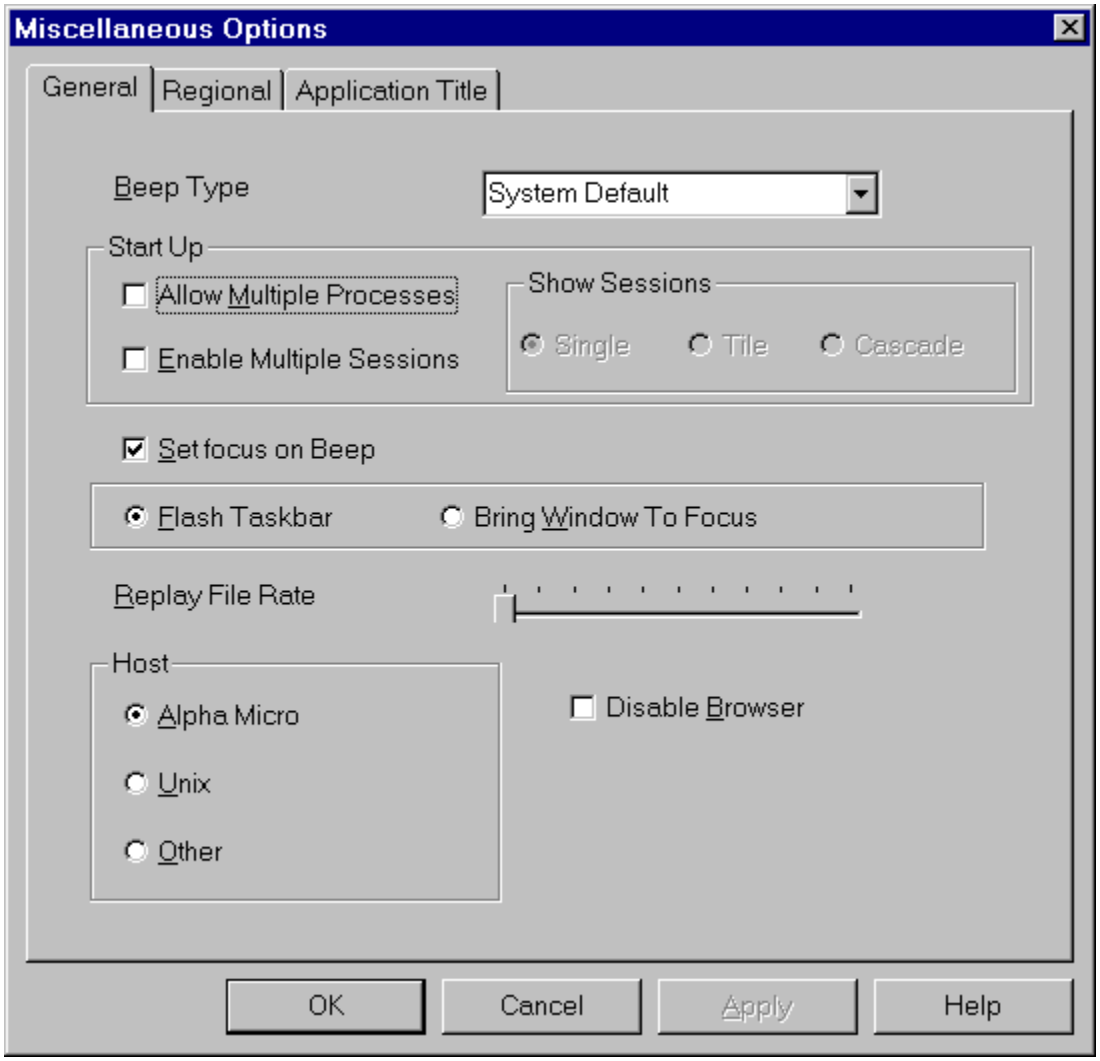
4.9 Starting Multiple Sessions or Multiple Processes

You can start AlphaLAN++ 8.0 in the following 3 modes.

4. **Single Process, Single-session mode (default)**. If the user double clicks on application-icon(s) to start it again while the application is still running, a new session or process would NOT start. This prevents users from accidentally opening up multiple application windows or sessions.
5. **Single Process, Multiple-session mode**. The user can double click on application-icon(s) to start another session (within the same application window). Please check the “Enable multiple sessions” box in the following window to select this mode. You can also select to start and show these multiple sessions in “single”(default) or “Tile or “Cascade styles. The user can switch between the

sessions by using Control+F10 key. If you want to automatically start multiple sessions, you may put multiple copies of the “Launch AlphaLAN++” icon in the start up folder.

6. **Multiple Process mode**. The user can double click on application-icons to start another application window. Please check the “Allow multiple processes” box in the following window to select this mode. This differs from the previous mode in the sense that you have two separate application windows. Windows task manager would show two different application EXE programs running. The user can no longer use the Control+F10 key to switch between these windows. The windows key, ALT+F10, would need to be used to switch from one application window to another. However if you have a large monitor and need to maintain different window positions and sizes for these application windows, you can do so. Please remember to start the application icons with different “configuration file (.aln) names”, if you wish the application to remember different window positions and sizes.



Disclaimer

The information in this document could include inaccuracies or typographical

errors. U.A. Systems, Inc. makes no representation or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, U.A. Systems, Inc. reserves the right to make changes from time to time without obligation of U.A. Systems, Inc. to notify any person or organization of such changes.

U.A. Systems, Inc.

1232 N. Monte Vista Ave, Unit #5
Upland, CA 91786

Voice: (909) 608-1575
Fax: (909) 608-1578
E-mail: support@uasys.com