WE-D320

Installation and
User Reference Manual
Version 1.63x

For NEXTSTEP

and X windows

workstation ag · Rohrstrasse 36/38 · CH-8152 Glattbrugg

The information in this manual is believed to be correct as of the date of publication, however it is subject to change without notice and does not represent a commitment on the part of workstation ag. workstation ag disclaims any warranty of any kind, whether express or implied, as to any matter whatsoever relating to this manual, including without limitation the merchantability or fitness for any direct, indirect, special, incidental or consequential damages arising out of purchase or use of this manual.

The software described in this manual is furnished under a license agreement and may only be installed, used or copied in accordance with the terms of that agreement.

workstation ag · Rohrstrasse 36/38 · CH-8152 Glattbrugg

Table of Contents

1 Preface

1.1 Purpose and audience

This manual allows UNIX workstations to interact with DEC™ host applications designed for DEC™ terminals running in VT320 (or VT100) mode. This is a user's guide, it is aimed at end users who are UNIX users and are familiar with using the DEC™ display terminal and printer. The typical audience is also familiar with DEC™ host login and the host applications accessible by the DEC™ display terminal product.

1.2 Summary of content

Chapter 2: "Overview"
presents an overview of the WE-D320 terminal emulation package. Included are possible configurations of the software package, supported features, and system software requirements.

Chapter 3: "Installing WE-D320"
describes the tape contents and installation procedure for WE-D320.

Chapter 4: "Configuring WE-D320"
describes the general considerations for configuring workstations and terminals.

Chapter 5: "Using WE-D320 terminal emulator"
explains how to use the emulation menus to make your screen customization. This chapter also describes the keyboard layout window, the print layout panel, the macro window and the color mapping window.

Chapter 6: "Keymapper tool"
explains how to use the keyboard mapping utility KM-D320 delivered with your emulation.

Appendix A: "WE-D320 parameter file format"
lists an example WE-D320 parameter file and explains each entry.

Appendix B: "Status line, window border line, icon and messages"
provides description of the WE-D320 messages areas and their respective content.

Appendix C: "Network WE-D320 installation"
a proposal for a WE-D320 network wide installation allowing any user to use the product the same way anywhere on the network.

Appendix D: "M_func_names" list
provides a list (and brief description) of all WE-D320 function names which can be mapped to the buttons placed around the window (see "Buttons_File" on page 66).

1.3 Conventions used

Throughout this manual we use the following conventions:

The *host system* is the DEC™ mainframe computer to which the WE-D320 device emulator communicates.

The *application program* is the job running on the host mainframe to and from which the emulated terminal operator sends and receives data.

1.4 Text fonts used

Throughout this manual fonts are used as follows:

`Typewriter font`
Represents what the system prints on your workstation screen, as well as UNIX system program names and files.

`Boldface typewriter font`
Indicates literal user input, typically commands and responses to prompts that you can type in exactly as printed in the manual.

Boldface font
Emphasized information within the text.

Italic font
Indicates variables or parameters that you replace with an appropriate word or string. Also used for emphasis.

Gray-shaded boxes 
Contain text that represents interactive sessions. User input is indicated by boldface typewriter font.

Non-shaded boxes 
Contain text that represents listing and non-interactive sessions.

1.5 References

If you need additional informations on any of the major topics mentioned but not explained in this manual, you might try these sources:

DEC™ (Digital Equipment Corporation) publications

1. EK-VT320-RM-001 VT320 Programmer Reference Manual
2. EK-VT320-UG-001 Installing & Using the VT320 Terminal

NeXT manuals

1. User's Reference Manual
2. System and Network Administration

Sun manuals

1. Beginner's Guide to the Sun Workstation
2. System and Network Administration

2 Overview

2.1 Introduction

The WE-D320 product line centers around three products that allow workstations to communicate with DEC™ mainframes over links provided from your workstation software:

#	WE-D320	Terminal emulation process
#	KM-D320	Interactive keyboard mapper
#	WE-D320_SERIAL	Serial link interface process

These products provide interactive access to the many applications that run on DEC™ mainframes without modifying the mainframe software.

2.2 Configurations

Since most workstations sold today offer TCP/IP, asynchronous serial ports and X25 PAD access software, WE-D320 will use these standard communication facilities.

Therefore, only the WE-D320 emulation component will be used in most cases. The user will start any process like a UNIX shell (sh, csh, ksh, etc.), a remote login (rlogin) or any other application in the terminal emulation window.

The WE-D320_SERIAL component may be used when the communication with the DEC™ mainframes is done over a serial line. Although “tip” (a standard process for accessing the serial ports) could be used, 8 bit data communication is not always possible. In those cases, WE-D320_SERIAL can be used as serial interface handler.

2.3 Feature summary

The WE-D320 terminal emulation offers support for applications written for the VT100, VT220 and VT320 DEC™ terminal families.

2.3.1 WE-D320 datastream operations

All operations accepted by the original DEC™ VT320 like:

Cursor positioning operations
Changing character sets
Status reports
Changing display format
Setting attribute combinations (including oversized characters, blink, etc.)

2.3.2 WE-D320 keyboard support

All functions of the WE-D320 keyboard are implemented directly on the workstation keyboard (configurable through an interactive keyboard mapper utility) or on the screen area around the terminal window (clickable with the mouse).

2.3.3 Limitations against DEC™ VT320

1. WE-D320 Release 1.6xx doesn't support downloadable character sets.
2. WE-D320 Release 1.6xx doesn't support VT52 mode.
3. Host application controlled print is not implemented in Release 1.63x.

2.3.4 Additional features not provided by the DEC™ VT320 terminals

– Unlimited number of simultaneous WE-D320 sessions (windows) on a single workstation. Possibility to save screen space by iconifying a window without interruption of the terminal session.

– Cut and paste between WE-D320 windows and other windows.

– Resizeable window width and height.

– Context sensitive help for all keys and parameters.

– Keyboard mapping and up to 12 macro functions can be defined separately for each WE-D320 window.

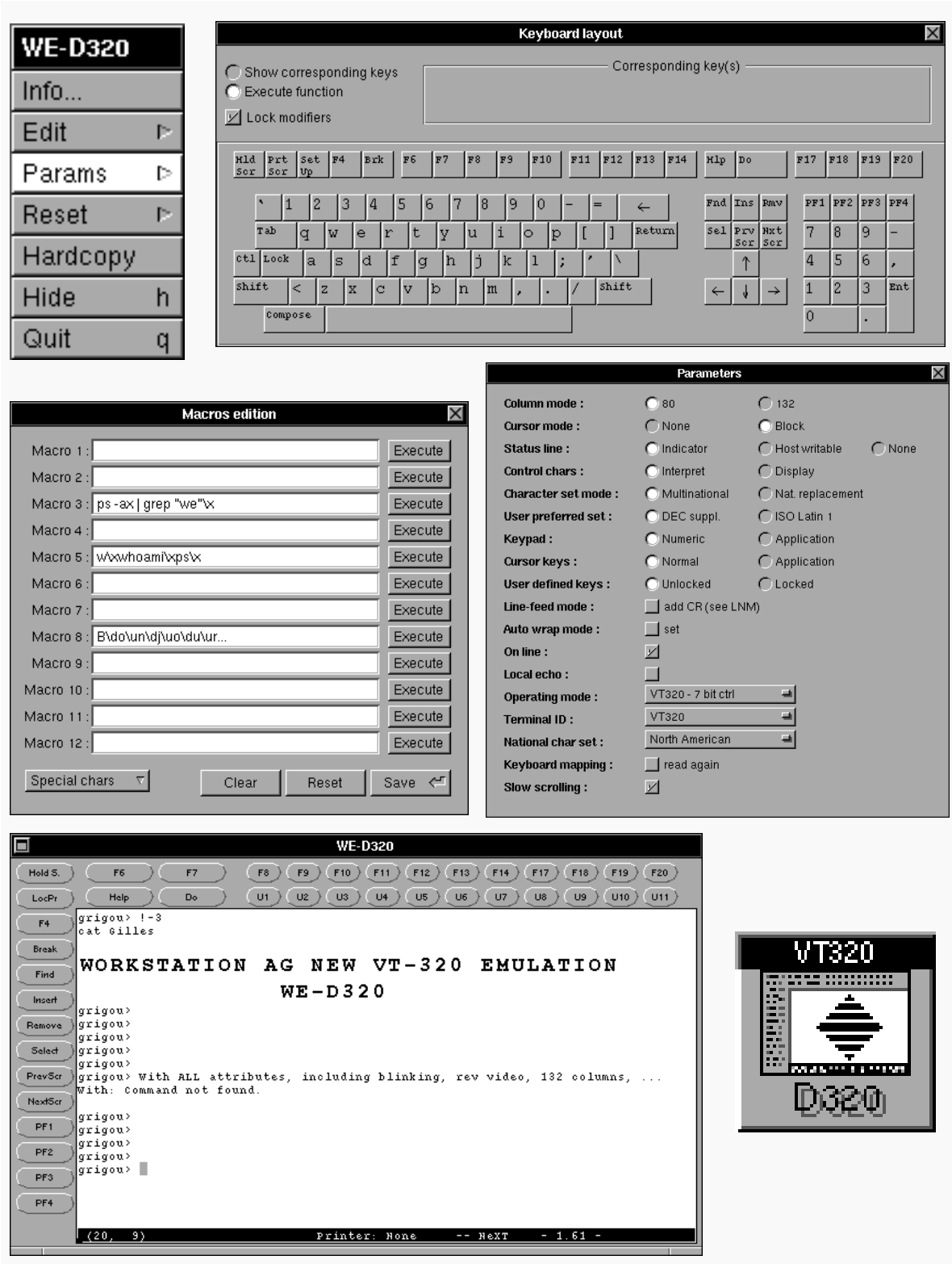
– A graphical representation of an original DEC™ VT320 keyboard can be displayed at any time on the screen and used with the mouse.

– Data destined to the printer (screen hardcopies or data coming directly from the host) are directed either to a file or to the printer spooler. This means that the host will never have to wait for printer readiness, paper low, etc. You can choose between ASCII and PostScript hardcopies.

NOTE: Host controlled print is not implemented in Release 1.63x.

Overview

The figure below shows you a couple of WE-D320 windows and panels as they appears on your workstation screen (NEXTSTEP example).



3 Installing WE-D320

3.1 Extracting the product from the distribution media

The procedure for extracting the products from the distribution media may vary depending on your particular workstation and/or environment. The actual procedure is explained on a sheet attached with your media. Please read this paper and follow the instructions contained therein.

When the installation is complete, you should find the following files in the installation directory:

`README`	The latest information about WE-D320
we-d320	The terminal emulator
we-d320_serial	A process to access serial lines
km-d320	The keymapping tool
Sample.D320_config	A sample configuration file for WE-D320
keylay.dat	An image of the DEC terminal keyboard
color.dat	A sample color file
macro.dat	A sample macro file

If you purchased WE-D320 for an X11 server machine, the following supplemental file will be provided:

ps_generic.dat	Used to generate PostScript hardcopies

3.1.1 Creating configuration file(s) for WE-D320

For a complete description of all possible entries in a parameter file, please refer to Appendix A on page 53. Thereafter, we will discuss where and how WE-D320 searches for parameter file(s) and some general rules valid for all entries in these files.

3.2 Format of an entry in a parameter file

All entries have the format:

ParameterName	*Value*	*Comment*
i.e.: `Term_Name`	`vt220`	`# Terminal type`

If you don't want to assign a new value to a parameter, you **must** remove the corresponding entry from the parameter file. A line like

```
Term_Name
```

with no *Value* would set `Term_Name` to **nothing**. This is surely **not** what you want ...

3.2.1 Up to four parameter files are possible

When started, WE-D320 looks sequentially for four parameter files (all have the format specified in Appendix A on page 53).

– First, it looks for a file with the name <**.we-d320-rc**> in the <**/etc/we-d320-config**> directory. If it is found, the parameters contained are extracted and validated.

– Secondly, it looks for a file with the name <**.we-d320-rc**> in the user's home directory. If it is found, the parameters contained are extracted and validated.

– Thirdly, the <command line> is searched for a <**-sf *FileName***> argument. If one is present, the parameters contained are extracted and validated.

– Fourthly, the <command line> is searched for a <***FileName***> entry as the first argument. If one is present, the parameters contained are extracted and validated.

NOTE 1: There is no obligation to have (or use) all four parameter files. This is provided to give you more flexibility to divide between common parameters for all WE-D320 sessions, less common entries valid for a group of sessions and finally parameters valid for a unique session only (i.e.: Term_Name).

NOTE 2: If any parameter is defined in more than one of the four possible files, the value read in the last file will be applied to that parameter. This allows you to apply your own default values for some parameters.

3.2.2 Installing the <WE-D320.info> file for On-line help

The <WE-D320.info> file provides help information for WE-D320. You may use it on any X window system. Here is how WE-D320 searches for this file:

– If HELPPATH is set in your environment, you merely need to copy the <WE-D320.info> file into the directory indicated by HELPPATH.

– If HELPPATH is not set in your environment, you may either:

  – Set HELPPATH and copy the <WE-D320.info> file there.
  – Copy the <WE-D320.info> file into </usr/lib/help> (default).

NOTE: There is currently **no** help available for the NEXTSTEP version.

4 Configuring WE-D320

4.1 Starting the "WE-D320" terminal emulator

The syntax for starting the "WE-D320" is as follows:

we-d320 [*FileName*][**-sf** *FileName*]

1. <'*FileName*'> File name of configuration file for WE-D320

2. <-sf '*FileName*'> Special configuration file for WE-D320

All the configuration files have the same format (see "Creating configuration file(s) for WE-D320" on page 13).

Example:

```
machine% we-d320 /usr/we-d320/john/myconfig1
           -sf /usr/we-d320/all/grpconfig
```

Then, "WE-D320" will try to read (in sequence):

– A file named <.we-d320-rc> in the </etc/we-d320-config> directory.

– A file named <.we-d320-rc> in your home directory.

– A file named <grpconfig> in the </usr/we-d320/all> directory.

– A file named <myconfig1> in the </usr/we-d320/john> directory.

4.1.1 Error messages while starting the "WE-D320" terminal emulator

The following are fatal errors which can occur when starting the WE-D320 terminal emulator. When they occur, **no** WE-D320 window will be created and you will have to correct the error(s) before proceeding.

a) If the password is missing or wrong, you will get a message or panel like:

```
WE-D320: Sorry, the password is corrupted:
```

b) If any of the font parameters from the configuration file(s) can't be found, you will get the following message:

```
WE-D320: Font <FontName> NOT found
```

Where: `<FontName>` is the name found in the configuration file(s).

Action => Specify fonts accessible to your X server. Don't forget that only fixed point fonts are usable for this terminal emulation.

4.2 Notes for NeXT users

Under NEXTSTEP, you may start an emulation by double-clicking on an icon corresponding to a configuration file. This is possible only if the two following conditions are met:

– The application (we-d320) is installed in a directory known by the Workspace Manager like:

  /Apps

  ~/Apps

  /LocalApps

  ...

– The configuration file you want to click on have the extension:

  .D320_config

If both conditions are met, the <xxx.D320_config> file will appear in your browser with the same icon as the <we-d320> application.

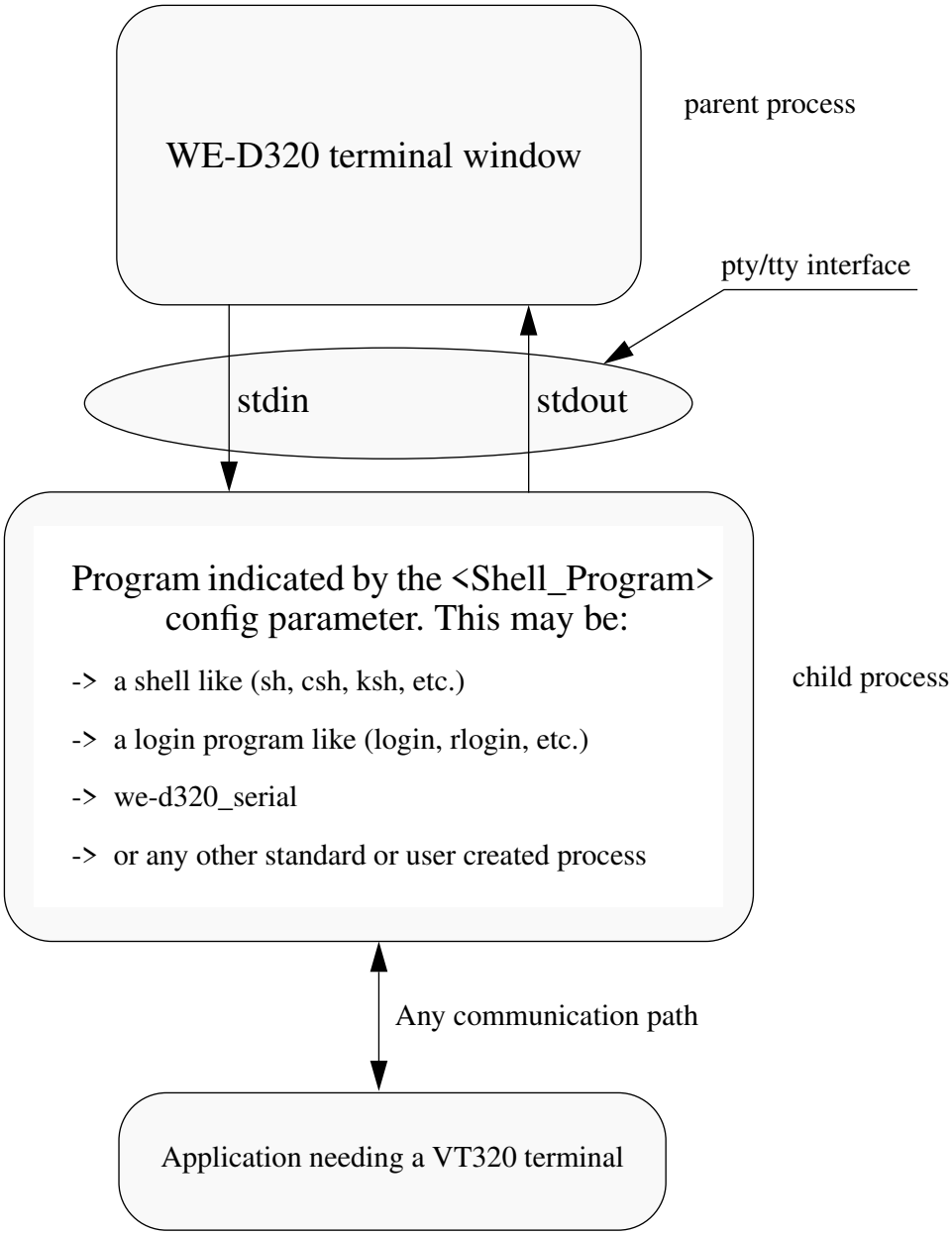
REMARK: During emulation start-up, some of the error messages described in the previous chapter may be replaced by panels for better readability. Other messages may be sent to the system console. Therefore, it is recommended to have a console window open during the emulation installation phase.

When started by double-clicking, <WE-D320> will still try to read

– A file named <.we-d320-rc> in the </etc/we-d320-config> directory
– A file named <.we-d320-rc> in your home directory

to get configuration information. The configuration file you click on will be read last and parameters contained herein may overwrite previous defined values.

4.3 How WE-D320 connects to an application through its child process



At start-up time, the WE-D320 emulator will start the process whose name has been found in the <Shell_Program> config entry as its “child process”.

NOTE 1: The WE-D320 parent process will arrange for being the process handling the child's standard input and output operations. Furthermore, due to the fact that a **pty/tty interface** has been setup by WE-D320, the child may use any UNIX system call (read, write, ioctl, fcntl, etc.) to set the interface parameters for its "stdin", "stdout" and "stderr" at will. Those features are used automatically by shells and many other standard programs you may start within your WE-D320 emulator window. These technical informations are for those who want to start a custom application within WE-D320.

NOTE 2: As you can see on the picture on page 17, WE-D320 doesn't connect directly to any communication path. This is done by the child process itself or by any other process started by this child. This will allow you to effectively use any communication path you may have installed on your workstation like:

Serial lines
An X25 PAD
Ethernet, Decnet, Netware or any other LAN or WAN
Program to program communication
etc.

4.4 Hints about locking configuration file(s)

If you intend to run many WE-D320 window on a single or on multiple workstations, you need control on the usage of some connection parameters.

For that reason, we have provided the <Lock_Setup> parameter in your configuration file(s). The configuration file, which will be locked if you set this option (set it to 1), is the one given without <-sf> as the 1st argument on the command line.

If you try to start a second window with a locked parameter file, you will get a message on your terminal or on the system console telling you, that the configuration file is locked and no window will be created. This also provides the possibility to create a configuration file management software examining which configuration(s) is (are) currently in use.

REMARK: The <Lock_Setup> option can only be used if you have <write> access to the concerned configuration file. If, this is not the case, an error message may be displayed on your terminal or on the system console.

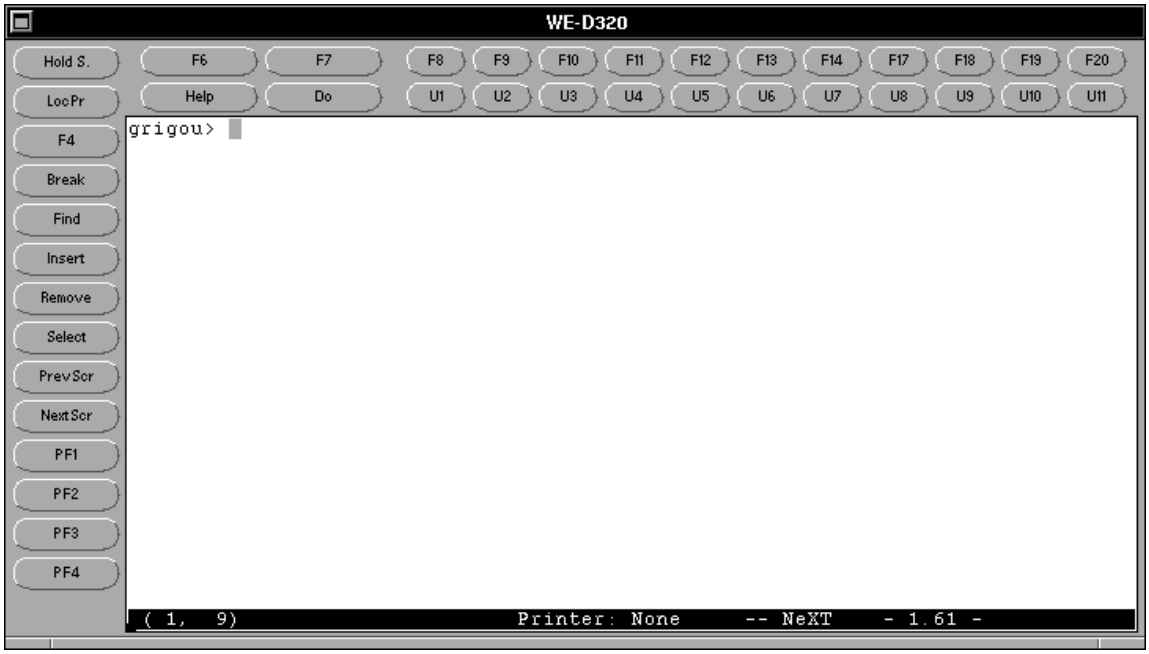
5 Using WE-D320 terminal emulator

5.1 Foreword

This chapter applies to the X and NEXTSTEP version. When nearly identical, only an X or a NEXTSTEP picture is shown in this manual. When differences exist, the X picture is shown on the left and the NEXTSTEP picture on the right.

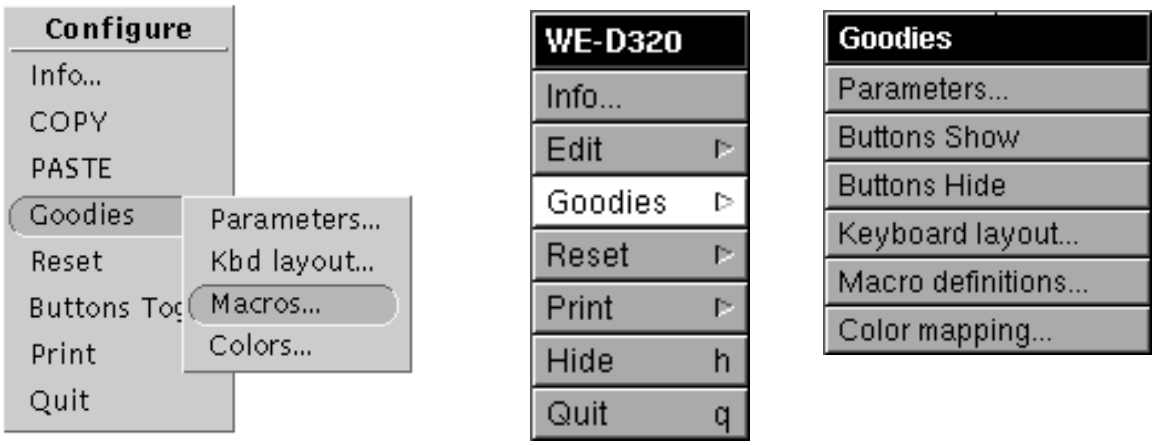
This chapter introduces you to the terminal emulator.

If you start the WE-D320 with a Cshell inside (the “Shell_Program” parameter is set to “csh”), the following window appears (data will probably be different):



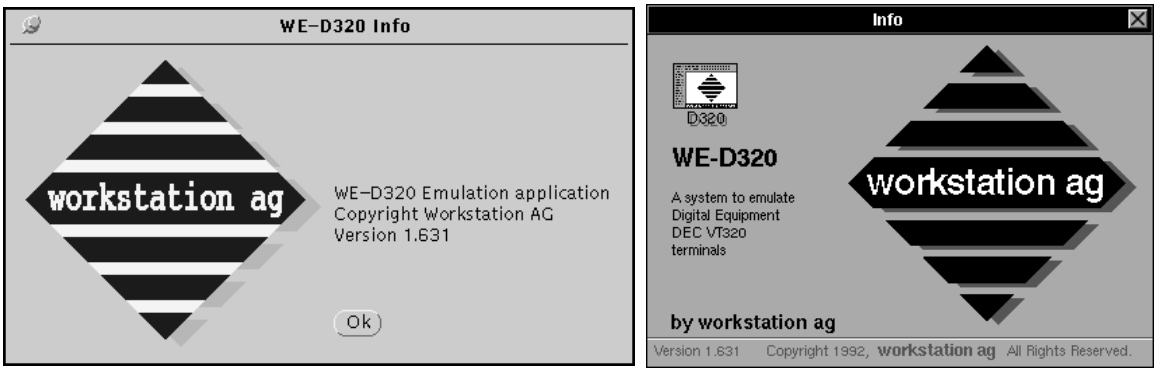
5.2 Main menu

The two pictures on page 20 show the Main menus for X and NEXTSTEP. Under X, you will get it by clicking the right mouse button, while the cursor is in the emulator main window. For NEXTSTEP, it will appear on the top left of your screen while the emulation window is selected.



5.2.1 Info

Will display an information panel containing the software release, copyright information, etc.



5.2.2 Copy, Paste

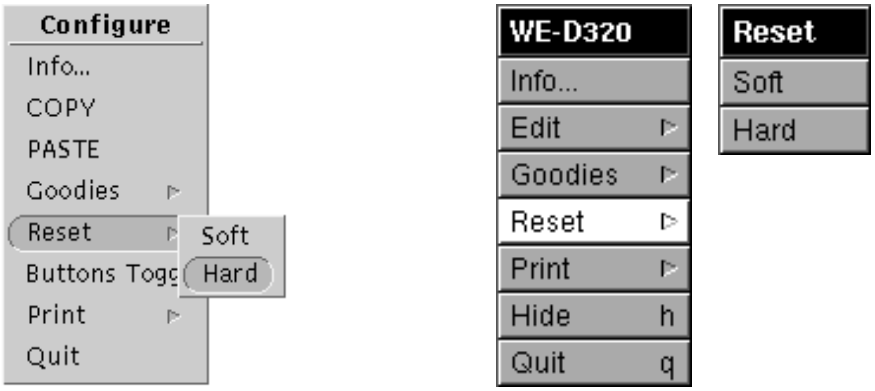
Will allow you to Copy and Paste between emulator windows and other applications. The way you can use Copy and Paste in the emulation is the same as for any other application.

NOTE 1: Under NEXTSTEP, Copy and Paste are in the **Edit** submenu.

NOTE 2: Under X, the Copy and Paste protocol used is ICCCM compliant. Since most X applications use this protocol. Copy and Paste will work between WE-D320 and most of your other applications.

5.2.3 Reset

Allows you to reset some emulation parameters and status. You may choose between two levels of reset:



Soft => Is equivalent to DEC™ VT320 "Soft Terminal Reset (STR)". This kind of reset has the following effect:

- Reset auto wrap
- Reset insert mode
- Unlock the keyboard
- Set the multinational set
- Set the numeric/keypad to numeric
- Enable the cursor
- Set normal graphic rendition

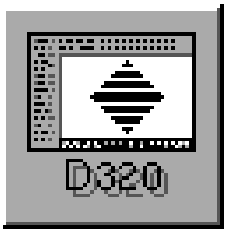
Hard => Is equivalent to DEC™ VT320 "Reset to Initial State (RIS)". This kind of reset has the following effect:

- Clear the screen
- Set screen format to 80 columns mode
- Reset all tabs

REMARK: Whenever possible, avoid using any reset function. If you are forced to do so (i.e.: for unlocking the keyboard), try with the soft reset function first. It should be sufficient in most cases.

5.2.4 Hide (NEXTSTEP only)

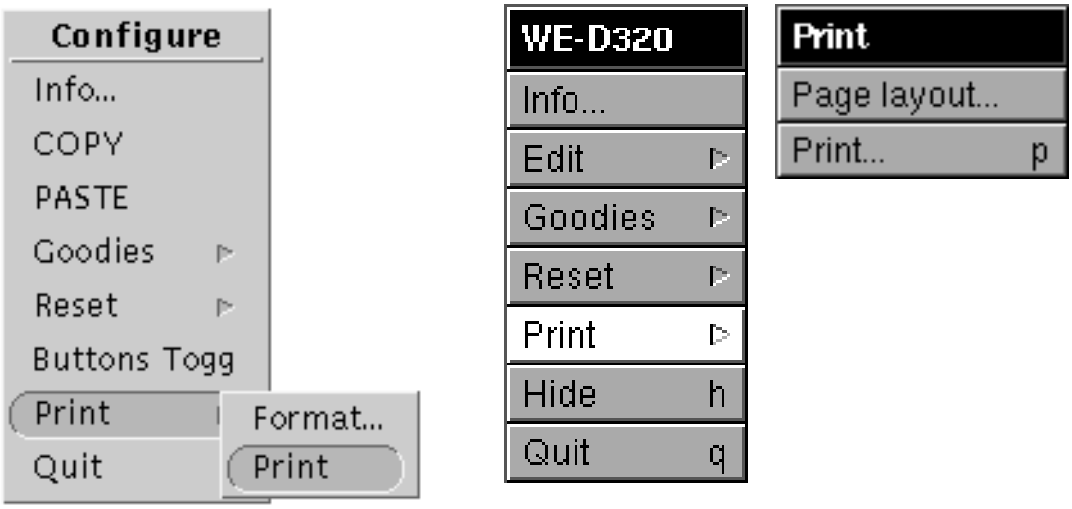
If clicked, any window or panel as well as the Main menu panel will be removed from your screen and only the WE-D320 icon will remain.



However, the aspect of the icon will always be the same. Therefore, we recommend **not** to use the Hide menu function but to iconify the emulation main window by clicking in the upper left button (see Appendix B on page 73).

5.2.5 Print

When you choose this option, the following submenu appears:



Print

Will make a hardcopy of the text window. The format of the printout is defined by:

-> Values set by the configuration file(s) read at WE-D320 start-up (see Appendix A on page 53).

-> Values entered at any time using the format or page layout panel.

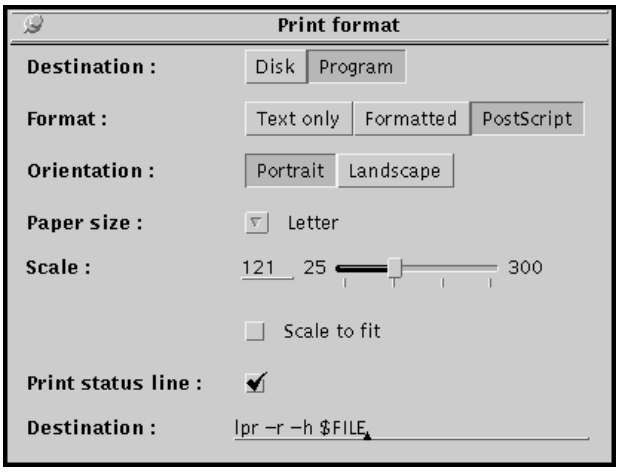
Format (X) **or page layout** (NEXTSTEP)

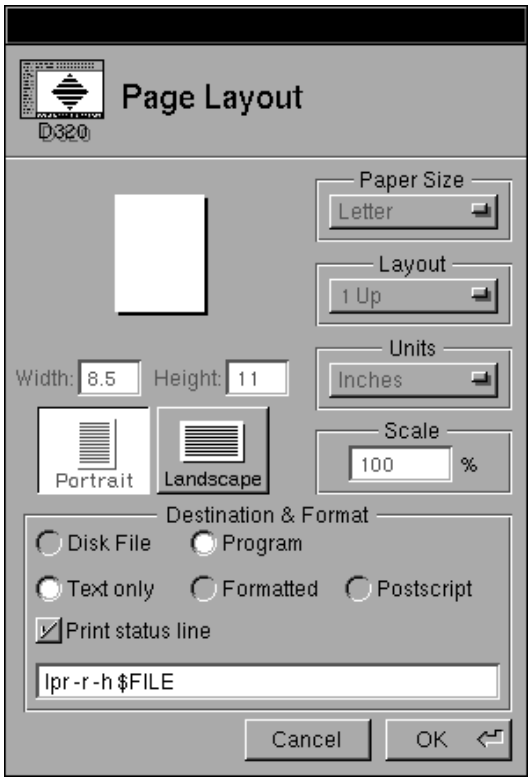
Foreword for NeXT users:

If you choose the PostScript “Print_Format” and make hardcopies, the standard NeXT print panel will appear each time as with all NeXT applications. With that panel, you will be able to print or save your print at will.

This means that the destination (disk file or program) and the text field at the bottom of the page layout panel entries are **useless** for NeXT PostScript hardcopies.

Clicking on format (X) or page layout (NEXTSTEP) will show one of the following panels:





At program start-up, these panels contain defaults or values set by the corresponding configuration (parameter) file(s) entries (see Appendix A on page 53).

Although the panels for X based machines and for NEXTSTEP looks very different. Buttons and text fields have the same function. Thereafter, we will discuss each of them:

Destination

-> Disk

WE-D320 will create disk files containing the print data in the given format. The text field at the panel bottom contains the file name. There are two cases:

– If you don't use $ext, the file name given will be unique and therefore be overwritten by each hardcopy (or data chunk coming from the host).

– If you put $ext at the end of your file name, new files will be created automatically for you by each hardcopy (or data chunk coming from the host). The following extensions will be added automatically by WE-D320.

If “Format” is Text only:

ProcessNumber-SequenceNumber-Kind

If “Format” is Formatted:

ProcessNumber-SequenceNumber-Kind.fmt

If “Format” is PostScript:

ProcessNumber-SequenceNumber-Kind.ps

Where:

– *ProcessNumber* is the UNIX allotted process number of the WE-D320 process.

– *SequenceNumber* is a number (starting at 0 and incremented by 1 at each print).

– *Kind* may be:

hcopy (for hardcopies)

CAUTION: If you edit the text field to change the file name, don’t forget to **press return** to validate the typed text.

REMARK: Defaults come from the following configuration file entries:

Print_Destination	*Disk*
Disk_Print	*~/MyPrints.$ext*

-> Program

WE-D320 will create temporary disk files (/tmp/we-d320/$ext) containing the print data in the given format for the meaning of $ext (see “Destination” on page 23). The text field at the panel bottom contains the name of the program to which WE-D320 will pass this temporary file for further handling. A typical entry in this text field may be:

lpr -r -h $File

where:

lpr	is the name of the program to be called.
-r -h	are parameters passed to program.
$File	token which will be replaced by WE-D320 with the name of the file just created.

CAUTION: If you edit the text field to change the program name, don’t forget to **press return** to validate the typed text.

REMARK: Defaults come from the following configuration file entries:

Print_Destination	*Program*
Disk_Print	*lpr -r -h $File*

Format

-> Text only

WE-D320 will create hardcopies containing exclusively standard ASCII characters. This means that attributes and foreign characters will not be rendered.

-> Formatted

WE-D320 will create hardcopies containing ASCII characters and the following DEC escape sequences:

ESC # (3 to 6)	To set character width and height
ESC [0 ; (1 to 2) m	To set highlight, underline, normal rendition

Choose formatted only if you have a printer (or driver) with the ability to handle DEC private escape sequences.

-> PostScript

WE-D320 will create hardcopies which will be an exact image of the current text window. If Print_Status_Line is **on**, the status line and a window border will be printed as well.

REMARK: Defaults come from the following configuration file entries:

Print_Format	*Text* *Formatted* *PostScript*

Orientation, paper size and scale

These entries are valid only for PostScript format. They are used by WE-D320 to generate the proper PostScript code according to the paper size, orientation and scale chosen.

REMARK: Defaults come from the following configuration file entries:

Print_Output_Size	*Letter* *A5* *A4* *A3*
Print_Orientation	*Portrait* *Landscape*
Print_Scale	*Any number between 30 and 300*

Print_Status_Line

If Print_Status_Line is **on** (set), the status line content will be added to hardcopies.

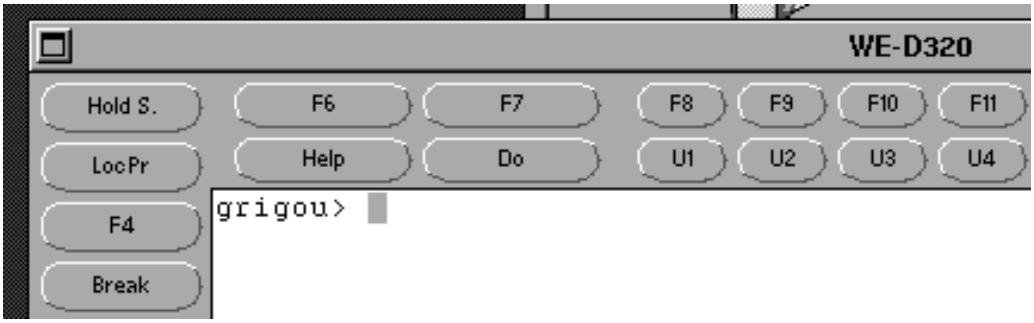
REMARK: Default comes from the following configuration file entries:

Print_Status_Line *0*

5.2.6 Buttons Show, Buttons Hide and Buttons Toggle

`Buttons Show`, `Buttons Hide` and `Buttons Toggle` allows you to switch between the window with buttons and without buttons.

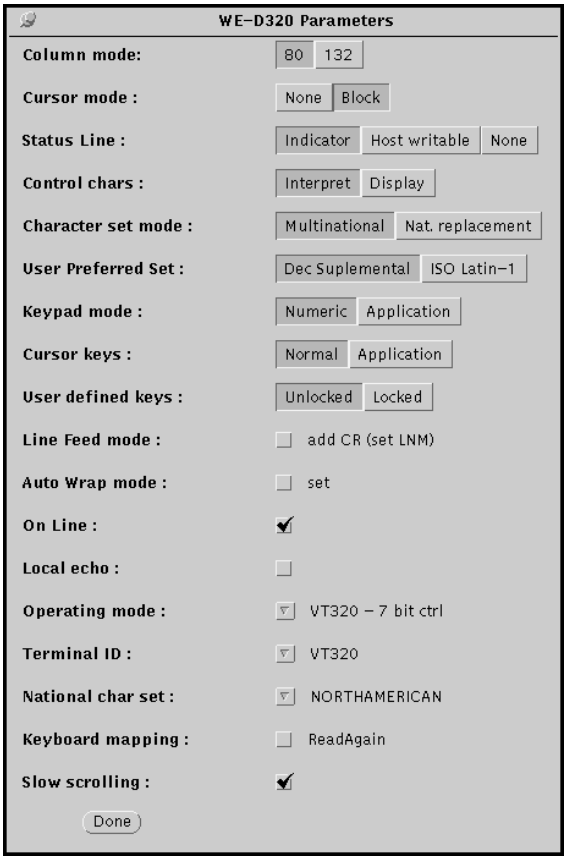
1. Example of window with buttons

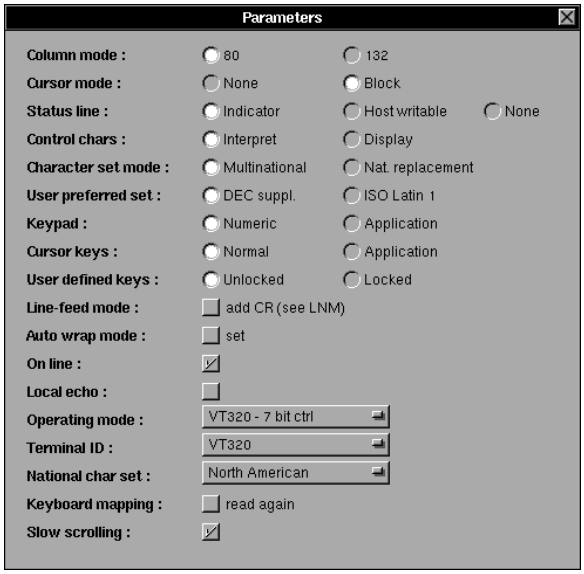


2. Example of window without buttons

5.2.7 Parameters

If you choose the parameter entry, the following window appears:





In the parameter window you can set all parameters described below. All parameters marked with an "*" may also be modified directly by the application running on the host computer you are connected to.

Column mode *

This field allows you to dynamically alter the number of characters per line.

-> 80: The window width will be set for 80 characters per line.

-> 132: The window width will be set for 132 characters per line.

The default will be either 80 or the <Screen_Cols> entry of your parameter file.

Cursor mode *

This field allows you to dynamically alter the cursor.

-> None: There will be no cursor displayed in the window.

-> Block: A block cursor will be displayed in the window.

The default will be either block or the <Cursor_Enable> entry of your parameter file.

Status line (last line of window) *

-> Indicator: The status line will display informations regarding the status of the emulation like:

– The cursor position
– The current machine and software release
– The printer status

For more informations, please refer to “Status_Line (* $)” on page 58.

-> Host writable: The status line can be used by the application to write application specific messages.

-> None: The status line is blank (no data displayed).

The default will be either indicator or the <Status_Line> entry of your parameter file.

Control chars

-> Interpret (default): WE-D320 will execute control characters (escape sequences) received from the application.

-> Display: WE-D320 will not execute but merely display control characters received (useful for debugging purposes only).

Character set mode *

Selects the type of character set used, multinational or national.

-> Multinational (default): Enables WE-D320 to generate 8 bit multinational characters, including 7 bit ASCII characters.

-> Nat. replacement: Causes WE-D320 to use one of eleven 7 bit national replacement character sets. The NRC set depends on the keyboard choused (see “National character set” on page 32).

User preferred set *

When *character set mode* (see page 28) is set to "multinational", selects the DEC multinational or ISO Latin-1 set.

-> DEC Supplemental (default): Selects the DEC multinational character set. This is compatible with Digital applications.

-> ISO Latin-1: Selects the ISO (International Standard Organization) character set.

NOTE: If your application can use the ISO set, you should select "ISO Latin-1".

Keypad mode *

Selects whether or not the keypad sends ASCII character codes or escapes sequences.

-> Numeric (default): ASCII character codes corresponding to the numeric characters on the keys will be sent.

-> Application: Escape sequences will be sent.

Cursor keys*

Selects whether the arrow keys send standard ANSI cursor control sequences or application specific control functions.

-> Normal (default): Arrow keys send standard ANSI cursor control sequences.

-> Application: Arrow keys send application specific control functions.

User defined keys

Selects whether or not the host can change user defined key (UDK) definitions.

-> Unlocked: Lets the host change UDK definitions.

-> Locked: Does not let the host change UDK definitions.

Line feed mode (LNM)

Selects the action taken when WE-D320 receives a LF, FF or VT. Selects whether or not WE-D320 will send CR or CR/LF when you press Enter.

-> Set: When WE-D320 receives a LF (Line Feed), FF (Form Feed) or VT (Vertical Tab.), it will also execute a CR (Carriage Return). When you press Enter, WE-D320 will send CR/LF.

-> Reset: When WE-D320 receives a LF, FF or VT, it will remain on the same column position as before. When you press Enter, CR alone will be sent.

Auto wrap mode

Selects whether or not display text automatically wraps on screen.

-> Auto wrap (set): Causes a character received after the right margin to automatically appear in the first character position of the next line.

-> No auto wrap: Causes characters received after the right margin to be overwritten into the last position of the current line.

On-line

Selects whether or not data are passed to the connected host (application).

-> On-line (set): Data typed will be passed to the connected host (application). This is the normal setting.

-> Off-line (reset): May be used to make some local experiments without disturbing the host (application).

Local echo

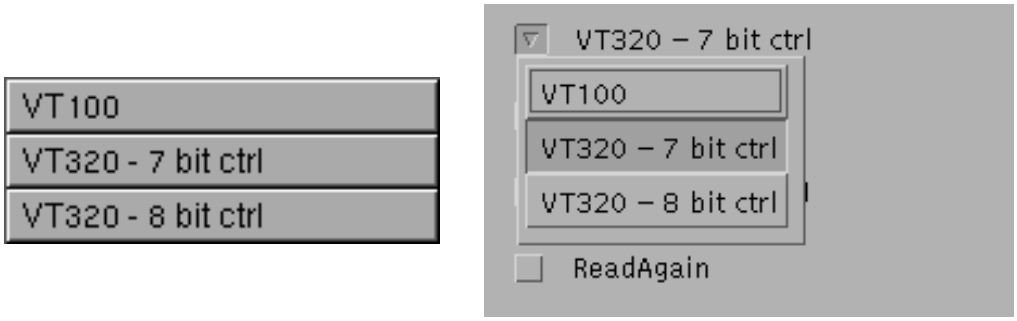
Selects whether or not data passed to the connected host (application) are also echoed locally to the WE-D320 screen.

-> Local echo (set): Data typed will be passed to the connected host (application) and echoed as well.

-> No local echo (reset) default: Data typed will be passed to the connected host (application) which may send an echo or not.

WARNING: Setting local echo may produce doubled characters in your terminal window.

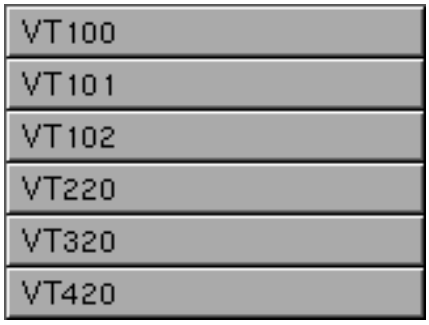
Operating mode

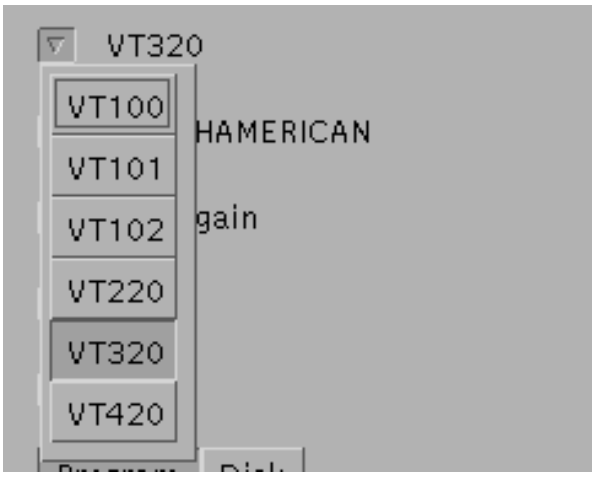


Selects the terminal operating mode.

-> VT100:	Lets the terminal run VT100 applications. In this mode, data and control characters both have 7 bits.
-> VT320-7 bit ctrl (default):	Lets the terminal run VT200/300 applications. In this mode, data characters have 8 bits and control characters have 7 bits. This is the mode recommended for most applications.
-> VT320-8 bit ctrl:	Lets the terminal run VT200/300 applications. In this mode, data and control characters both have 8 bits. This is the most efficient mode, but not yet supported by many applications.

Terminal ID





Selects the terminal device attribute response, also called terminal ID. This ID is sent in response to specific host application request.

National character set

Since WE-D320 uses the keymapping tool to map physical keys to graphics or functions, the setting you choose in the above menu will only be used to choose between the 11 national replacement character sets (NRC's).

-> North American: National replacement mode is disabled.

-> All others: Will be used to select 1 of 11 national replacement character sets (NRC's) while working in national replacement mode" (see "Character set mode *" on page 28).

Keyboard mapping

This entry allows you to reread the keyboard mapping file without leaving the WE-D320 emulation. The "Keymap_File" is reread when you click ReadAgain. This is mostly used at installation time to test a newly created keyboard mapping file. Information about the keyboard mapper are found in chapter 6 on page 43.

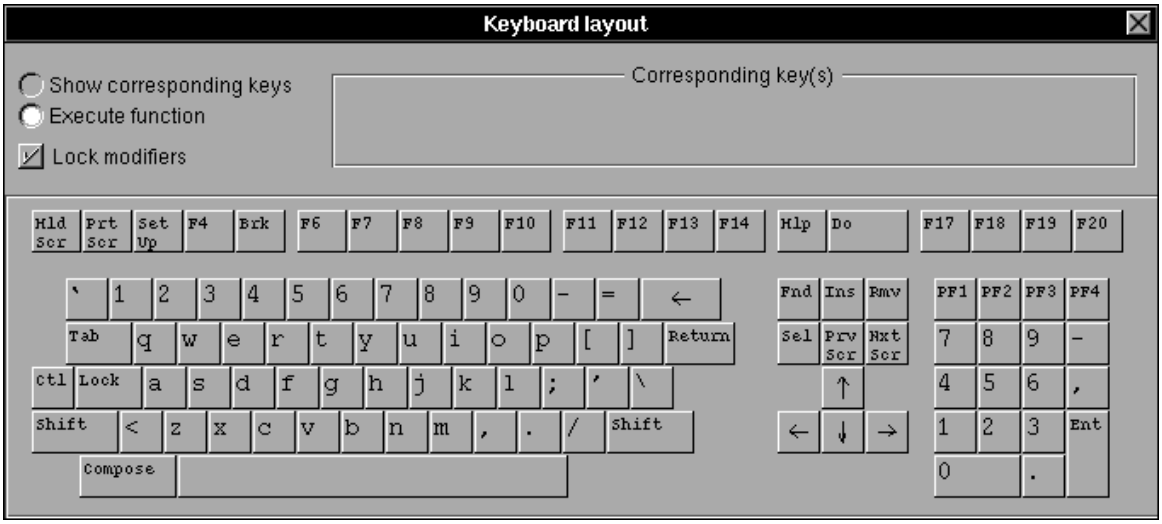
Slow scrolling

– If you set this entry, data coming from the host will be displayed line by line, thus enhancing readability at the cost of slightly degraded performance.

– If this entry is not set, data coming from the host will be displayed as fast as possible. This mode may produce a multiple record transmission data display.

5.2.8 Keyboard layout

If you choose the keyboard layout entry, the following window appears:



This is a picture representing an original VT320 keyboard. The layout shown may change according to your choice in the “National character set” entry on page 32. Like on the original, you may enter data, move the cursor, transmit data and so on. Because you have only one mouse (but ten fingers) and some functions or characters may only be produced by depressing several keys simultaneously, we introduced the

Lock modifiers

Button, which allows you to lock the special keys (will highlight when locked) and change the meaning of some keyboard areas accordingly. This is the same principle as (Shift Lock) on a typewriter. To unlock a modifier key, simply click on the highlighted key a second time.

Execute function mode

When this mode is active, any key clicked with your mouse will enter a character into the emulator window or execute some function (i.e.: Return, Tab, F6, etc.). Beside your workstation’s physical keyboard and the buttons around the emulator window, this is a third way to enter data into your WE-D320 emulation window.

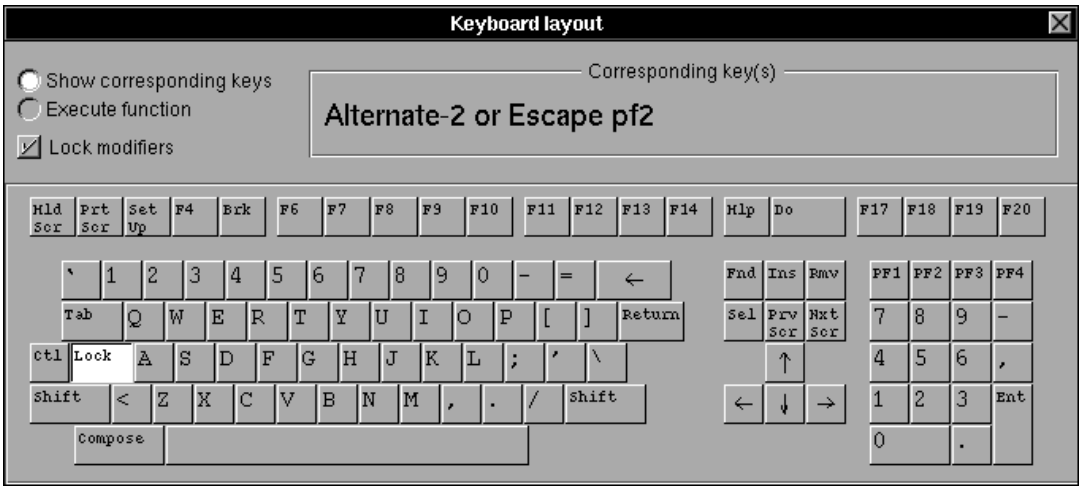
Show corresponding keys mode

When this mode is active, any key clicked with your mouse will display the actual key combination you should type on the physical keyboard of your workstation to obtain the same function as on the original VT320 keyboard displayed here.

The assignment of a key, a combination or a sequence of keys to a particular VT320 function is done with the ***keymapper tool*** described in chapter 6 on page 43. The show mode described here has been created to help you remembering the assignments you have done.

REMARK: When you click on the keyboard layout in show mode, either the ***key combination*** or the ***comment*** you have entered for this particular combination is shown (see picture below). Because the key combination can be somewhat cryptic (machine understandable only!), we warmly recommend that you enter comments which actually describe the key combination in an understandable way.

If a comment is available for an entry, it will always be displayed by the emulator.



This picture shows you a typical show mode output. The clicked on PF2 and the following line was displayed:

Alternate-2 or Escape pf2

=> This is the comment field which was entered with the ***keymapper tool***. It means that the user may produce the PF2 function by pressing either *Alternate-2* or the *escape introducer key followed by the text pf2.*

NOTE 1: In fact, when the user presses “Alternate-2” together, some code (i.e.: “Meta-2”) will be sent from your window server to the WE-D320 emulation program which, in turn, will send the same character string as PF2 of the original DEC™ VT320 keyboard to the host application.

NOTE 2: The keyboard mapping can’t be modified from within the emulation. If you want to modify your key assignments, change or add comments, we recommend the following procedure:

How to modify your keyboard mapping

1. Look in your configuration file for the name of the keyboard mapping file (the <Keymap_File> entry). For our example, we will assume that its name is <km.dat>.

2. Start an emulation with this configuration file and connect to an application using the key you want to map.

3. Start a keymapper tool (KM-D320) with the same keymap file (km.dat) as parameter.

4. Make any changes you want using the "keymapper tool". When you are done, don't exit the keymapper tool but use the ***Save*** button to save your changes to the keymap file.

5. Go back into the emulation, display the parameter (Goodies) panel and click the ***ReadAgain*** button. This will read your just saved <km.dat> into the current emulation session and make it immediately active.

6. Try your new key combinations within the emulation. If you are happy with your changes, you may now terminate the "keymapper" process. Otherwise, you may repeat steps 4 to 6 as many times as you want.

Common problems and errors

– When you enter comments in the "keymapper tool", don't forget to press the return key to terminate it. Otherwise, the comment will be discarded.

– In step 4, don't forget to save before proceeding to step 5 and activating ReadAgain. Otherwise, you will read the old version of your keymapping and **no** change will occur.

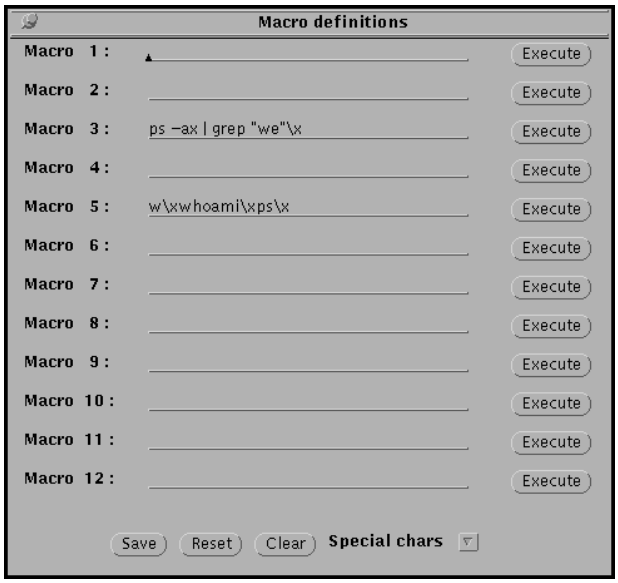
– In step 4, you should comment all entries with a sentence stating clearly what keys are to be depressed to produce the corresponding VT320 function.

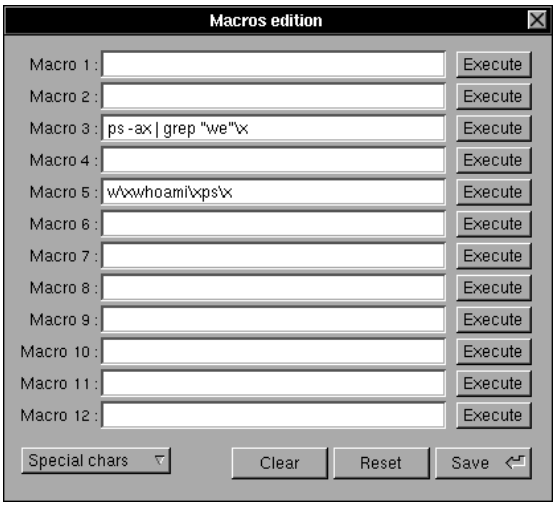
– If you don't know anymore which key combination you have assigned to a VT320 function, display the "keyboard layout" (Goodies), and use it in show mode.

NOTE: For more informations about keyboard mapping, please refer to chapter 6 on page 43.

5.2.9 Macro definitions

If you choose the macro definitions entry, the following window appears:





Purpose of this panel is to allow you to examine, execute or modify your current macro (key sequences) assignments. The actual content of the entries has been read from the <Macro_File> entry of your current configuration.

Clear button

When clicked, this will clear **all** macro entries. If you clicked it by mistake, you still can retrieve your macro entries by clicking the Reset button (see below).

Reset button

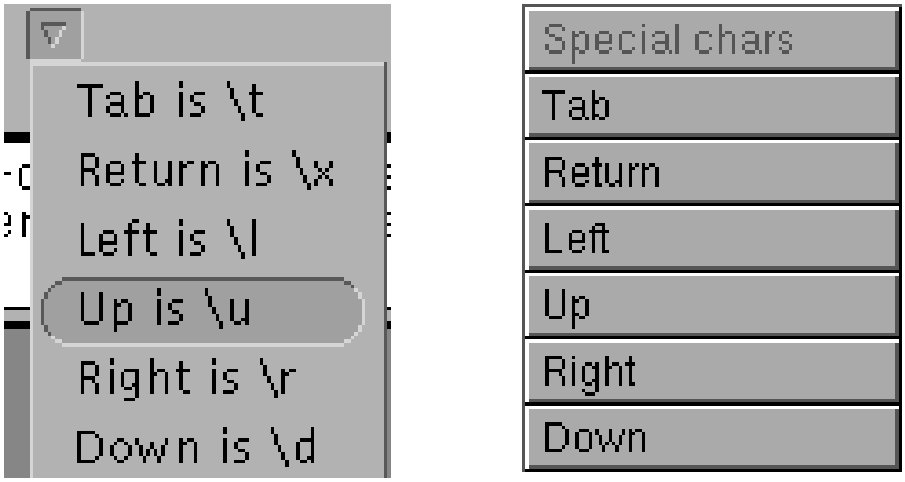
When clicked, this will revert **all** macro entries to the last saved version.

Save button

When clicked, this will save your current entries to disk. The file in which these macro entries will be saved is indicated by the <Macro_File> entry of your current configuration.

Special chars

When clicked, the following menu will be displayed:



Here is how to proceed

Let's say you want that macro 3 (when activated) puts ***ps -ax | grep "we"*** at the current cursor position and then sends *Return*.

1. Click at the beginning of the macro 3 field with the mouse.
2. Enter ***ps -ax | grep "we"*** in the macro 3 entry.

   For NEXTSTEP:
   With the mouse, click on special chars and select the **"Return"** entry. When selected, release the mouse button. This will automatically enter a "***\x***" entry in the macro 3 field.

   For X windows:
   With the mouse, click on special chars and read the indication on the right of the **"Return is"** entry. Then, enter the corresponding characters (i.e.: \x) into the macro 3 field.
3. Try this macro by clicking on the Execute button sitting on the right of the macro 3 entry.
4. Repeat steps 1 to 3 for each macro you want to (re)define.
5. When you are done, don't forget to click the **"Save"** button to write your macros on the disk.

REMARK: 12 keys (or key combinations) may be assigned for the execution of macros without displaying the present panel. For that purpose, you must use the "Keymapper tool" on page 43 and enter key combinations for the M_MACRO1EXEC to M_MACRO12EXEC entries.

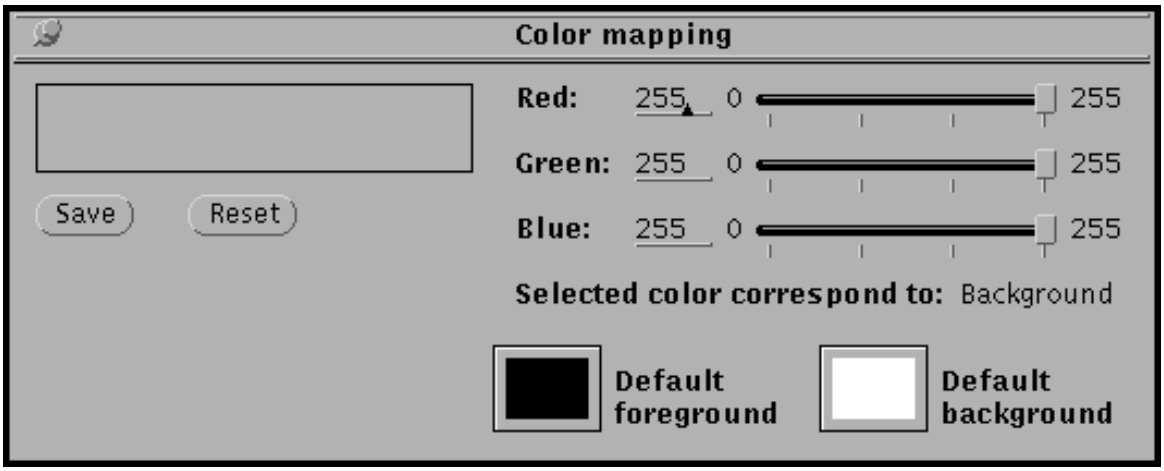
5.2.10 Color mapping

Purpose of the color mapping utility is to allow the user to edit and change the background and foreground colors of the WE-D320 window.

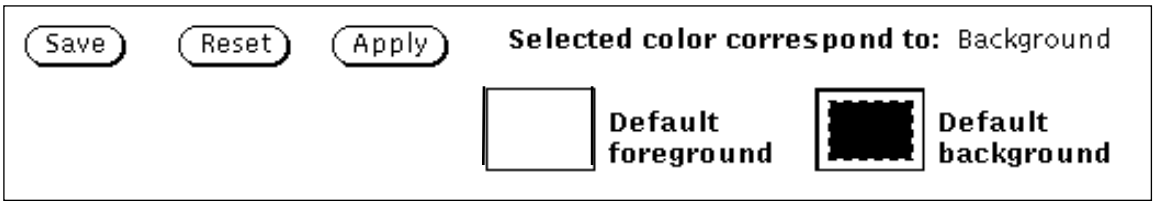
At emulation start-up, the color mapping comes from the <Color_File> entry of the configuration file. This file will be modified when you click the Save (X) or OK (NEXTSTEP) button of the color mapping panel.

5.2.11 Color mapping under X

When you select the color mapping entry of the Goodies menu, the following window appears:



If your X server has a Static Color Map, the lower part of the color mapping panel will look as below:



WE-D320 will switch automatically between Dynamic and Static Color Maps. If supported by your X server, Dynamic Color Maps will be used.

Reset button

When you click this button, the colors are reset to the values of the last saved version.

Save button

This button allow you to save your changes to disk. This action can't be undone.

How to proceed

1. Connect to the host application and display some data.
2. In the Goodies menu, select color mapping. The color editing tool panel (see the two pictures on page 38) will appear.
3. Within the panel, click on the color you want to modify. The name of the color you clicked on will appear in the panel (i.e.: background).
4. Use the three cursors to modify the color:

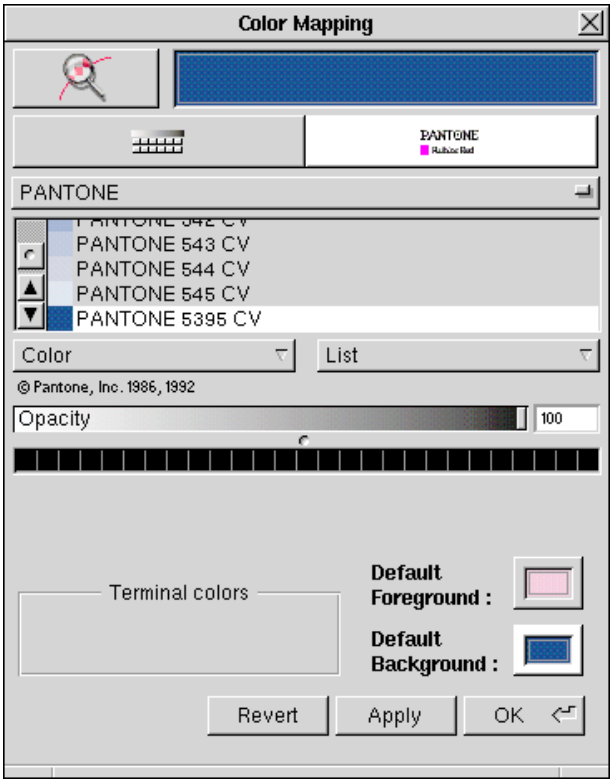
   -> For *Dynamic Color Maps*, the result will be immediately visible in the color mapping tool and in your emulator character window while moving the slider.

   -> For *Static Color Maps*, the result will be visible only in the color mapping tool after you have released the slider. To make the change appear in the emulator window, click the supplementary **Apply button**.

5. When you are done, click the Save button to save your work in the file indicated with <Color_File> in the active configuration (i.e.: color.dat).

5.2.12 Color mapping under NEXTSTEP

When you select the color mapping entry of the Goodies menu, the following window appears (NEXTSTEP 3.0):



REMARK: WE-D320 uses a NeXT object for the color edition. This object may be improved from one NEXTSTEP release to another and WE-D320 will benefit of those enhancements. Therefore, we will not describe the object itself, but only its relations with our particular application. For more informations about the NeXT color object and how to use it, please refer to your NeXT “User’s Reference Manual”.

Apply button

Click on this button to make your color changes appear in your emulator character (graphic) window without leaving the color mapping tool.

Revert button

Click on this button to revert the colors to what they where when you started the emulator.

OK button

Click on this button to exit the color mapping tool. Your changes will be saved to disk and applied to your emulator window (as with the Apply button).

How to proceed

1. Connect to the host application and display some data.

2. In the Goodies menu, select color mapping. The color editing tool panel (see picture on page 40) will appear (the panel aspect may change depending on your NEXTSTEP release).

3. Create your custom color the same manner you would do in any other NEXTSTEP application. When you are done, drag this color into the “Default Foreground” or “Default Background” color. The color will change to the one you dragged in. Click the Apply button to reflect this change in your emulator window. Repeat this step until you are satisfied with all colors.

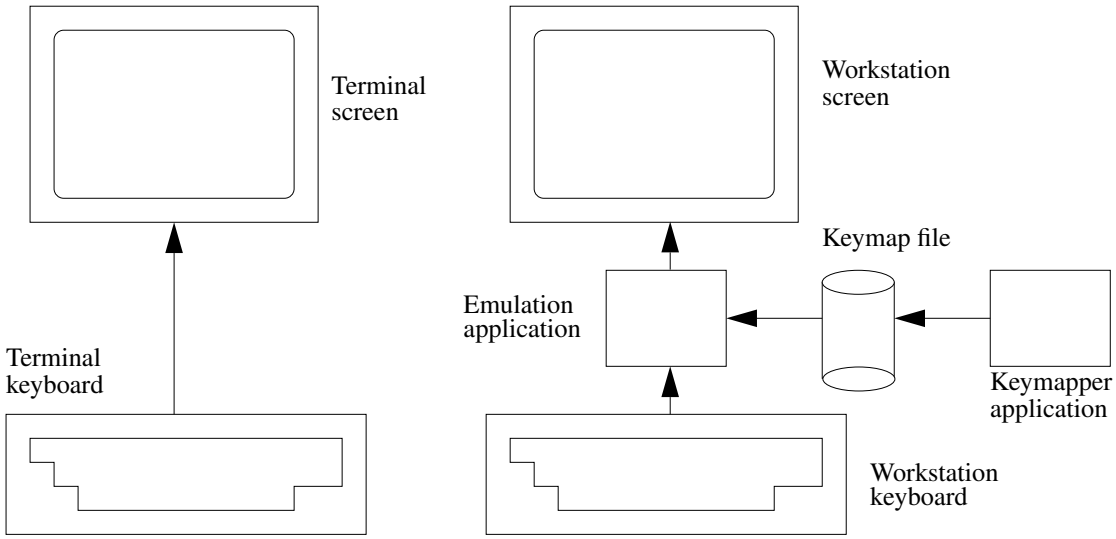
4. When you are done, click the OK button to save your work in the file indicated with <Color_File> in the active configuration (i.e.: color.dat). The color mapper tool panel will be closed.

5. If you want to reset the colors to what they were at emulation start-up time, you may click the Revert button. All your changes will be discarded.

6 Keymapper tool

6.1 Introduction

The keymapper is a tool application to perform keyboard mapping operations. The keyboard mapping is to assign some or all keys of the original terminal keyboard to your workstation keyboard. It is particularly useful to enter original terminal “keys”, that have no correspondence on your workstation keyboard.



Keyboard mapping isn’t integrated in the terminal emulation application because it is intended to be used by the system manager. Communication between the keymapper and the terminal emulation application is done through a file, the keyboard mapping file. This file contains associations between keys or key combinations and terminal keys or functions. The main purpose of the keymapper application is to provide an easy way for interactively building or editing the keyboard mapping file.

6.2 Definitions

Before describing further how to work with the keymapper application, some definitions must be stated:

6.2.1 A terminal function

is a function provided by the original terminal (and by the terminal emulation). Terminal functions can be (but aren’t necessarily) obtainable via the original terminal keyboard. Example of terminal functions (keys are enclosed in quotes): “Return”, “Backspace”, “Next Field”.

6.2.2 Terminal keys

are alphabetical, numeric and other ASCII symbol keys.

6.2.3 A modifier key

is a key like shift or ctrl. Its purpose is to change the code associated to a set of keys. Modifiers known by the application are shift, ctrl, meta (alternate on some keyboards) and numeric lock (num-lock). It is possible that you cannot access all of these modifiers with your workstation keyboard.

6.2.4 A key combination

is a single key typed with or without simultaneously stroking a modifier key. Example of key combinations (keys are enclosed in quotes): “a”, “A”, “Ctrl”-“c”, “Shift”-“Meta”-“F1”. You can notice that “A” is not, but could be “Shift”-“a”: every workstation company has its own policy for assigning codes to its keyboard(s) keys, that is why a keyboard mapping tool is useful.

6.2.5 An escape sequence

are keys typed one after the other (without modifiers) proceeded by a dedicated key combination (can have modifiers) called the **escape sequence introducer key combination** (or more quickly the **sequence intro key**). The purpose of escape sequences is to increase the possibilities of function mapping when the workstation keyboard offers too few possible key combinations. Example of escape sequences (the escape intro key is “Meta”-“e”, keys are enclosed in quotes, sequences are separated by commas): “Meta”-“e” “f” “c”, “Meta”-“e” “F6”, “Meta”-“e” “t” “r” “y”.

NOTE: For VT320 emulation users:
Escape sequences can replace the “Compose” key mechanism. You can assign all compose keys to escape sequences.

6.2.6 A special key

is a key that doesn’t generate an alphanumerical character. Special keys are keys used to do formatting (carriage return, tabulations, moving cursor), function keys or diacritical signed characters (é, ^, Ç, etc.). Special keys have names, always starting with a slash (‘/’); they are listed in chapter 6.7 on page 50.

6.2.7 An exotic key

is a key that doesn’t have a corresponding name. An exotic key is handled with an immediate (cryptic) value instead of a name. Exotic keys should be rare on the standard keyboards.

6.3 Installing and starting the keymapper tool

6.3.1 Foreword

This chapter is common for the whole WE-XXX product family. Presently, this includes the

WE-UTSc / WE-UTSg / WE-D320 / WE-I3179c / WE-I3179g

products. Since all pictures were done with the *WE-UTSc* product, the illustrations shown may be somewhat different for other products. However, all buttons, functions and procedures are common to all products.

6.3.2 Under NEXTSTEP

Copy the application into the “LocalApps” or “NeXTApps” folder, or in the “Apps” folder below your home folder. Copy the keyboard mapping file anywhere you want. Copy the keyboard layout file into your home folder (if you want to start the application by a double-click on its icon) or in the current directory (if you want to start the application through a shell window). The keyboard layout file must be named “keylay.dat” or the “-klf” command line option should be used (for more see “Tips for advanced users” on page 49).

To be recognized as the document of the keymapper application (visualized with the same icon than the application), the keymapping file must have the “*xxx*_keymap” extension (where *xxx* is the name of the terminal emulation <UTS, D320, I3179>). Like every NeXT application, the application must reside in one of the standards application folders.

6.3.3 Under X windows

Copy all the files from the distribution media into a directory, and type in a shell window:

```
cd program_directory
km-xxx km.dat
```

NOTE: Depending on the particular WE products, “km.xxx” can be one of the following:

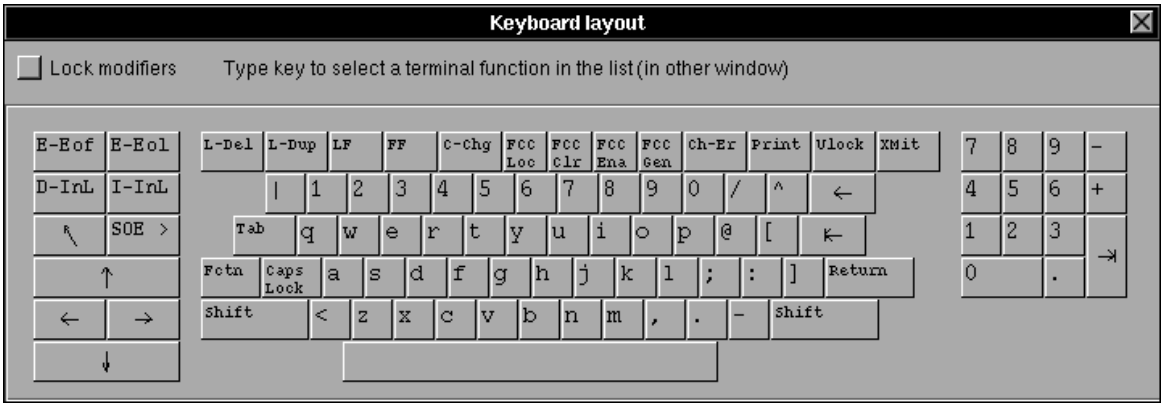
km-uts	for the WE-UTS emulation
km-d320	for the WE-D320 emulation
km-i3179	for the WE-I3179 emulation

Then, proceed with the next chapter.

6.4 Using the keymapper tool

6.4.1 Usage overview

To start the keymapper application, type its name in a command shell or double-click on the keymapping file icon. After a few seconds, two windows should be visible: the first containing an image of the original terminal keyboard, the other containing a list of terminal functions and some buttons, this is the keymapper main window.



To make associations between original keyboard functions and key combinations of your physical workstation keyboard, proceed as follows:

1. Click any key of the keyboard layout window: you can see that the corresponding terminal function (or key) is selected.

2. Now select the main window and type any key or key combination on your workstation physical keyboard. You can see that the "name" of this key is displayed at the left of the terminal function list: you have created an association between the selected terminal function and the displayed key combination.

NOTE 1: In step 1, you could also directly select a terminal function by choosing an item of the terminal function list.

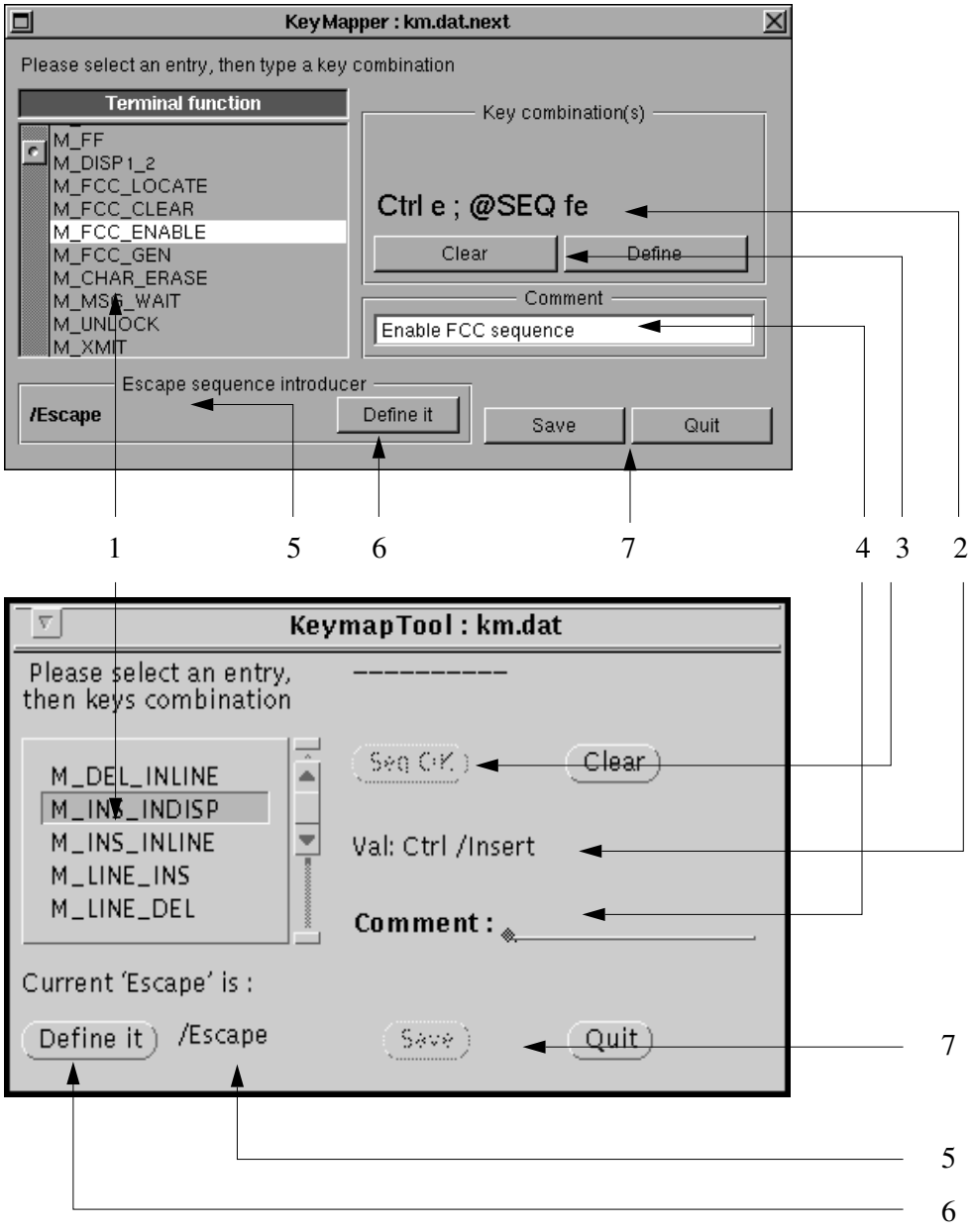
NOTE 2: (For NeXT users only): You cannot use the "Command" modifier in the key combinations you define, because "Command" should be used only for menu shortcuts (accelerators), and terminal functions are not provided in the application menus.

The illustrations on page 47 show the main window of the keymapper tool.

(1) The list of the terminal functions and keys. All terminal functions are proceeded by "M_" to avoid confusion with keys.

(2) The key combination associated to the currently selected terminal function (or key).

(3) The Define and Clear current key combination buttons.



(4) The comment associated to the currently selected terminal function (or key). You can edit this comment.

(5) The current escape sequence introducer key.

(6) The escape sequence introducer key definition button. After clicking this button you can define a new escape sequence introducer key by typing a key combination.

(7) The Save and Quit buttons.

6.4.2 Terminal function list

You can select a terminal function by clicking on an item of the list and browse through the list by clicking on the arrow buttons. You can notice that when selecting an item, the key combination field and comment change accordingly.

The list begins with the terminal functions (preceded by the “M_” sequence), continues with all the regular ASCII characters and terminates with the extended international characters.

NOTE: By default (in the delivered “km.dat” file), only the regular ASCII characters are mapped to their equivalent.

6.4.3 Key combination field

A terminal function or key can be associated to one or several key combination or escape sequences. If there are several key combinations (or escape sequences) they are separated by semicolon “;”. The modifier names are displayed in extension, the key name is either a letter or a string, for special keys (see chapter 6.7 on page 50). The sequence intro key is indicated with the “@SEQ” string. Exotic keys are introduced with the “Hexa” string followed by a number in hexadecimal base.

The application denies associating a key combination to two different terminal functions or keys. It shows a warning panel on the screen and cancels the entered key combination. You must remove the key combination associated to the other terminal function to associate it to the currently selected terminal function. Similarly you cannot define an escape sequence that is already included in another one: for example, the “@SEQ fe” cannot be defined if the “@SEQ fec” already exists.

6.4.4 Notes for NeXT users

– There is no difference between the numeric keypad and the numeric keys on the main keypad, because the NEXTSTEP environment returns the same codes for both key sets.

– You may get some strange results when typing key combinations. For example, if you type the “2” key and simultaneously press the “Ctrl” modifier, “Ctrl” will be displayed in the key combination field.

In fact, NEXTSTEP does part of the job of the keymapper application: when using key combination with modifiers (shift, alternate, ctrl) it returns the corresponding code. For example, when typing “Ctrl”-“A”, NEXTSTEP doesn’t return “A” with modifier “Ctrl” but the code of the control-A character (in our case ASCII code 01) with “Ctrl” modifier.

This behaviour can be handled by the keymapper application for “Ctrl” modifier on alphabetical keys, but with “Ctrl” modifier on other keys, and especially for “Alternate” modifier there is no way to retrieve the key combination from the code returned by NEXTSTEP (e.g. there is no way to retrieve the typed “Meta-s” from the received “Meta-ß” [“Meta /ssharp”] because on another NeXT keyboard [on a keyboard for a different country, for example] the “Meta-s” can generate another code).

6.4.5 Define and Clear buttons

These buttons allow you to change the key combination associated with a terminal function or key. Hit the “Clear” button to wipe-out all the previously defined key combinations of the currently selected terminal function. Check the “Define” (“SEQ end”) button to notify the end of the escape sequence currently being defined.

6.4.6 Comment field

To edit a comment, click in the comment field, type your new comment and don’t forget to **press the return key** to indicate that you’ve finished editing a comment (otherwise you won’t be able to enter new key combinations; they will accumulate in the comment field).

NOTE: We warmly recommend that you enter comments which actually describe the key combination (or sequence) you have pressed to produce the codes. For example, if you pressed “Alternate-t” to be assigned to, let’s say home, a possible comment would be:

For home press <Alt and t> simultaneously.

The comment field gets displayed in the <keyboard layout> window of the emulation when set in <show mode>. The user may read it by clicking on a key in the <keyboard layout> window (Goodies menu) while in show mode.

6.4.7 Escape sequence intro key

To edit or set the escape sequence introducer key you must click the “Define” button and then type the key combination you want. This will change the sequence intro key for all future and already defined escape sequences.

NOTE: You may define only one sequence intro key.

6.4.8 Save and Quit buttons

The “Save” button allows you to save the current keyboard mapping definitions. The “Quit” button is for quitting the application: a panel is prompted, providing a last chance to save the modifications you made to the definitions, if any.

6.5 Tips for advanced users

6.5.1 Command line parameters

There are some options that you can add to the command line when starting the application. They allow you to customize the behaviour of the application.

-nc *national_charset_name*

Forces the application to display the given national variant of the original terminal keyboard. This affects the keyboard layout window. The different national character set names are defined in the emulation documentation, for the same option (“-nc”). The default national variant used is the US one.

-lnr *number_of_entries*

Specifies the number of entries that can be simultaneously displayed in the terminal function list (affect the size [in pixels] of the scrolling list). Setting the number too high (typically > 20) will have an unpredictable effect.

-klf *name_of_keyboard_layout_file*

Specifies the name and the path of the keyboard layout file, the file describing the layout of the terminal's original keyboard. The default name of the keyboard's layout file is "keylay.dat". The default path is the current directory.

-flk *name of font*

Allows to change the font used for the labels of the keyboard layout keys. The font name must honour the syntax used by display environment (X or NEXTSTEP). Choosing an unappropriate font may cause problems (partially hidden key labels). Default font used is courier. For X, this font must be available at least in 10, 12 and 14 points size.

6.6 Editing the keyboard mapping file

There is a possibility to manually edit the keyboard mapping file. However, you will loose some of the checks made in the keymapper application and there is no guarantee that the key combinations you define can be generated by your workstation keyboard.

The syntax used in the keymapping file is described in the file itself.

6.7 Names of the special keys

Return	Return key
/Backspace	Delete previous character
/Tab	Forward tabulation
/RevTab	Backward tabulation (NeXT only)
/Up, /Left, /Right, /Down	Arrow keys (for moving cursor)
/Escape	Escape key
/KP_*x*	Numeric keypad keys (*x* is a digit or other symbol)
/F*x*	Function keys
/*x*acute	(*x* is a vowel) á, é, í, ó, ú
/*X*acute	(*X* is a capital vowel) Á, É, Í, Ó, Ú
/*x*grave	(*x* is a vowel) à, è, ì, ò, ù
/*X*grave	(*X* is a capital vowel) À, È, Ì, Ò, Ù

/*x*diaeresis	(*x* is a vowel) ä, ë, ï, ö , ü, ÿ
/*X*diaeresis	(*X* is a capital vowel) Ä, Ë, Ï, Ö, Ü, Ÿ
/*x*circumflex	(*x* is a vowel) â, ê, î, ô, û
/*X*circumflex	(*X* is a capital vowel) Â, Ê, Î, Ô, Û
/aring, /Aring	å, Å
/oslash, /Oslash	ø, Ø
/ntilde, /Ntilde	ñ, Ñ
/otilde, /Otilde	õ, Õ
/ccedilla, /Ccedilla	ç, Ç
/ae, /AE	æ, Æ
/eth, /Eth	
/thorn, /Thorn	
/ssharp	ß
/exclamdown	¡
/cent	¢
/sterling	£
/currency	¤
/yen	¥
/brokenbar	
/section	§
/diaeresis	¨
/copyright	©
/ordfeminine	ª
/guillemotleft	«
/notsign	¬
/hyphen	–
/registered	®
/macron	¯

/degree	°
/plusminus	±
/onesuperior	
/twosuperior	
/threesuperior	
/acute	´
/mu	µ
/paragraph	¶
/periodcentered	·
/cedilla	¸
/masculine	º
/guillemotright	»
/onequarter	
/onehalf	
/threequarters	
/questiondown	¿
/multiply	×
/divide	÷

A WE-D320 parameter file format

```
# WE-D320 WINDOW PARAMETERS
Term_Name          Vt220          # Name of terminal for "termcap" or "terminfo"
Screen_Rows        24             # WE-D320 window will have 24 rows
Screen_Cols        80             # and 80 columns
Buttons_Panel      1              # The window will be bordered with buttons
Slow_Scroll        1              # Better for your eyes
Exit_Confirm       1              # Avoids unwanted exits. Brings a warning panel
Lock_Setup         0              # The primary configuration file will NOT be locked

# WE-D320 BLINKING OPTIONS
Blink_Cursor       1              # Cursor will blink
Blink_OutFocus     0              # Blinking will occur only while window selected
Inactiv_Time       15             # Any blink activity will stop after 15 seconds

# THOSE MAY BE CHANGED BY THE HOST APPLICATION AS WELL
Cursor_Enable      1              # Yes, we want a cursor displayed
Status_Line        Indicator      # Format of 25th line = status
New_Line           0              # Return will send CR alone, not CR/LF
Local_Echo         0              # No, wait for application echoing
Auto_Wrap          1              # Go to next line automatically
Nrc_Mode           0              # Not in national replacement character set mode
Vt_Mode            VT320_8        # Will start in 8 bit control mode
User_Set           DEC            # DEC multinational set
Keypad_Mode        Numeric        # Keypad keys send numeric characters
Cursor_Keys        Cursor         # Cursor keys send ANSI sequences

# THOSE MAY NOT BE CHANGED BY THE HOST APPLICATION
Keyboard_Type      NORTHAMERICAN  # US NRC set will be used
Udk_Lock           0              # Host can change UDK's
Term_Id            VT320          # We will respond as being a VT320
On_Line            1              # Yes, we want to talk to DEC

# WE-D320 FONTS
Font_R             -b&h-*-medium-r-*-*-%d-*-*-*-m-*-iso8859-1
Font_B             -b&h-*-bold-r-*-*-%d-*-*-*-m-*-iso8859-1
Font_Dimensions    8,10,12,14,18,19,24  # Useful for X11 based servers only
Start_Height       12             # Font height at start-up time will be 12
Font_P             -b&h-*-medium-r-*-*-%d-*-*-*-m-*-iso8859-1
Font_Warning       1              # Will print warnings for font problems
Invert_Bold        0              # Don't permute normal and bold attributes
```

```
# WE-D320 CHILD PROGRAM
Shell_Program       csh             # Mostly sh, csh, ksh, rlogin, tip, etc.

# WE-D320 AUXILIARY FILES
Keymap_File         km.dat          # Your keyboard mapping
Macro_File          macro.dat       # Your favorite macros
Kbd_Layout_File     keylay.dat      # The original DEC keyboard
Color_File          color.dat       # Your favorite colors
Buttons_File        button.dat      # Your custom buttons definitions

# WE-D320 PRINT PARAMETERS
Print_Format        Postscript      # Hardcopy output in PostScript format
Print_Status_Line   1               # Add status line and frame to output

# The following useful only if "Print_Format" IS NOT "PostScript"
Disk_Print          ~/we-d320.$ext  # File to print on
Print_Program       lpr -r -h $File # Print program to fork
Print_Dest          Disk            # Where print data will go

# The following useful only if "Print_Format" IS "PostScript"
Print_Output_Size   A 4             # Generate PostScript for A4 paper
Print_Orientation   Landscape       # Mostly the best solution
Print_Scale         100             # A 1/1 representation
Print_Generic_File  ps_generic.dat  # PostScript print template for X servers

# WE-D320 PASSWORD
Pass_Word           PutYourVendorSuppliedPasswordHere
```

We will now discuss each parameters in detail and specify whether they are mandatory, optional or if they have default values.

NOTE: All entries must be in the parameter file format (see page 13).

A.1 Parameter description

REMARK: The entries marked with an ***** can be modified at run time by clicking in the WE-D320 parameter panel (Main menu, Goodies entry) or in the print layout panel (Main menu, Print entry).

The entries marked with an **$** can be modified at run time by the host application to which you are connected.

Some entries are marked with both * and $. This means that both the host application and the operator using the WE-D320 parameter panel may modify the entries.

WE-D320 WINDOW PARAMETERS

1. Term_Name

– Definition: This parameter specifies the terminal type which will be set in the UNIX environment for this terminal window. This is used by many UNIX programs like “csh”, “vi”, etc. to make a search in the “termcap” or “terminfo” database. Possible values are:

Vt100
Vt220
Vt320
...

– Defaults: Vt320

– Mandatory: No

2. Screen_Rows

– Definition: This parameter specifies how many rows your WE-D320 window will have.
Possible values range from 24 to 48 rows.

– Defaults: 24

– Mandatory: No

3. Screen_Cols (* $)

– Definition: This parameter specifies how many columns your WE-D320 window will have.
Possible values are 80 and 132.

– Defaults: 80

– Mandatory: No

– Parameter panel: Column mode

4. Buttons_Panel (*)

– Definition: If set to 1 (YES), clickable buttons representing the special keys of an WE-D320 terminal (i.e.: function keys) will be displayed around the WE-D320 terminal window. You will then be able to activate all these special functions by clicking them with the mouse. Possible values are:

1 => buttons
0 => No buttons

– Defaults: 0 (No buttons)

– Mandatory: No

NOTE: At any time during the WE-D320 session you can toggle this option using the WE-D320 window menu. This option defines only the start-up value for this parameter.

5. Slow_Scroll (*)

– Definition: If you set this parameter to one, each line coming from the host application will be displayed separately, thus enhancing readability. The expense is an higher CPU load. Possible values are:

1 => Slow scrolling
0 => Scroll at highest possible speed

– Defaults: 0 (Highest speed)

– Mandatory: No

6. Exit_Confirm

– Definition: If you set this entry to 1 (YES), you will get a panel when you choose the <exit> or <quit> entry in the application menu. This will give you a chance to continue your session. Possible values are:

1 => Display a panel
0 => Exit immediately

– Defaults: 0 (Exit immediately)

– Mandatory: No

7. Lock_Setup

– Definition: If set to 1 (YES), the primary setup file (see “Hints about locking configuration file(s)” on page 18) will be locked (reserved) for the whole duration of the WE-D320 session. You may use this option only if you have write access to the primary setup file. Possible values are:

1 => Lock the primary setup file
0 => Do not lock the primary setup file

– Defaults: 0 (No lock)

– Mandatory: No

WE-D320 BLINKING OPTIONS

8. Blink_Cursor

– Definition: If set to 1 (YES), the cursor will blink. Otherwise, it will be a solid black rectangle. Note that the cursor may stop to blink when the WE-D320 window is out of focus (see "Blink_OutFocus" below). Possible values are:

1 => Blinking cursor
0 => Steady cursor

– Defaults: 0 (Steady cursor)

– Mandatory: No

9. Blink_OutFocus

– Definition: If set to 1 (YES), blinking fields will continue to blink in a non active window (out of focus). If set to 0 (NO), they will stop to blink thus lowering the processor load for that inactive window. Possible values are:

1 => Will continue to blink while window unselected
0 => Will stop blinking immediately

– Defaults: 1 (Continue to blink while not selected).

– Mandatory: No

10. Inactiv_Time

– Definition: This option gives the number of seconds after which any blink activity (cursor, blinking fields, etc.) will stop in case of inactivity. This will be effective even if the window is "In Focus" (is the active window). Values may range from 1 to 255 seconds. 0 means infinite time.

– Defaults: 0 (Infinite)

– Mandatory: No

PARAMETERS THAT CAN ALSO BE CHANGED BY THE HOST APPLICATION

11. Cursor_Enable (* $)

– Definition: With this option, you can decide wether or not a cursor will be displayed. We recommend to set this option to 1 (YES).

– Defaults: 1 (YES)

– Mandatory: No

– Parameter panel: Cursor mode

12. Status_Line (* $)

– Definition: At the bottom of the WE-D320 window, a supplementary line (mostly the line 25) exists. With this option, you can assign a function to this line. There are three possible options:

Indicator
The supplementary line will be used by WE-D320 to display informations useful for the operator. For a description of the different fields displayed in this indicator line, please refer to Appendix B on page 73.

Hostwrite
The supplementary line will be under control of the host application which will determine its content.

Nostatus
The supplementary line will be blank.

– Defaults: Indicator

– Mandatory: No

– Parameter panel: Status line

13. New_Line (* $)

– Definition: Selects the action taken when WE-D320 receives a LF, FF or VT. Selects whether or not WE-D320 will send CR or CR/LF when you press Enter. Possible values are:

1 (Set)
When WE-D320 receives a LF, FF or VT, it will also execute a CR. When you press Enter, WE-D320 will send CR/LF.

0 (Reset)
When WE-D320 receives a LF, FF or VT, it will remain on the same column position as before. When you press Enter, CR alone will be sent.

– Defaults: 0 (Reset)

– Mandatory: No

– Parameter panel: Line feed mode

14. Local_Echo (*)

– Definition: Selects wether or not characters typed will be echoed to itself by WE-D320. Possible values are:

1 (Set)
When you type characters, they are sent to the host application and echoed locally to the WE-D320 window. If the host application sends an echo, you may get the typed character twice on the screen.

0 (Reset)
Characters typed are only sent to the host application without local echoing.

– Defaults: 0 (Reset)

– Mandatory: No

– Parameter panel: Local echo

15. Auto_Wrap (* $)

– Definition: Selects whether or not display text automatically wraps on screen. Possible values are:

1 (Set)
Causes a character received after the right margin to automatically appear in the first character position of the next line.

0 (Reset)
Causes characters received after the right margin to be overwritten into the last position of the current line. Possible values are:

1 => Wrap mode active
0 => No wrap

– Defaults: 0 (No wrapping)

– Mandatory: No

– Parameter panel: Auto wrap mode

16. Nrc_Mode (* $)

– Definition: Selects if the national replacement character sets (NRC's) must be used:

1 (Set)
Causes WE-D320 to use one of eleven 7 bit national replacement character sets. The NRC set depends on the *Keyboard_Type* entry on page 62.

0 (Reset)
We-D320 will use multinational character sets. The multinational character set used depends on the *User_Set* entry below.

– Defaults: 0 (Multinational character sets)

– Mandatory: No

– Parameter panel: Character set mode

17. Vt_Mode (* $)

– Definition: Selects the terminal operating mode as follows:

VT320_8
Causes WE-D320 to start in VT320 8 bit ctrl mode.

VT320_7
Causes WE-D320 to start in VT320 7 bit ctrl mode.

VT100
Causes WE-D320 to start in VT100 7 bit ctrl mode.

– Defaults: VT320_7

– Mandatory: No

– Parameter panel: Operating mode

18. User_Set (* $)

– Definition: When *Nrc_Mode* (see above) is set to 0 (multinational), selects the DEC multinational or ISO Latin-1 set:

DEC multinational
Selects the DEC multinational character set. This is compatible with Digital applications.

ISO Latin
Selects the ISO (International Standard Organization) character set.

NOTE: If your application can use the ISO set, you should select "ISO Latin-1".

– Defaults: DEC multinational

– Mandatory: No

– Parameter panel: User preferred set

19. Keypad_Mode (* $)

– Definition: Selects whether or not the keypad sends ASCII character codes or escape sequences.

Numeric
ASCII character codes corresponding to the numeric characters on the keys will be sent.

Application
Escape sequences will be sent.

– Defaults: Numeric

– Mandatory: No

– Parameter panel: Keypad

20. Cursor_Keys (* $)

– Definition: Selects whether the arrow keys send standard ANSI cursor control sequences or application specific control functions.

Cursor
Arrow keys send standard ANSI cursor control sequences.

Application
Arrow keys send application specific control functions.

– Defaults: Normal

– Mandatory: No

– Parameter panel: Cursor keys

PARAMETERS THAT CAN'T BE CHANGED BY THE HOST APPLICATION

21. Keyboard_Type (*)

– Definition: Since WE-D320 uses the keymapping tool to map physical keys to graphics or functions, the setting you choose in the above menu will only be used to choose between the 11 national replacement sets (NCR's) while in this mode. A list of the possible entries follows:

-> NORTHAMERICAN
-> BRITISH
-> FLEMISH
-> CANADIANFRENCH
-> DANISH
-> FINNISH
-> GERMAN
-> DUTCH
-> ITALIAN
-> SWISSFRENCH
-> SWISSGERMAN
-> SWEDISH
-> NORWEGIAN
-> FRENCHBELGIAN
-> PORTUGUESE

– Defaults: NORTHAMERICAN

– Mandatory: No

– Parameter panel: National character set

22. Udk_Lock (*)

– Definition: Selects whether or not the host can change user defined key (UDK) definitions. Possible values are:

1 => Lock UDK's
0 => Do not lock UDK's

– Defaults: 0 (Unlocked)

– Mandatory: No

– Parameter panel: User defined keys

23. Term_Id (*)

– Definition: Selects the terminal device attribute response, also called terminal ID. This ID is sent in response to specific host application request. Possible values are:

-> VT320
-> VT220
-> VT102
-> VT101
-> VT100

– Defaults: VT320

– Mandatory: No

– Parameter panel: Terminal ID

24. On_Line (*)

– Definition: Selects whether or not data are passed to the connected host (application). Possible values are:

1 => On-line
0 => Off-line

– Defaults: 1 (On-line)

– Mandatory: No

– Parameter panel: On-line

WE-D320 FONTS

25. Font_R and Font_B

– Definition: These entries define the fonts to be used for the text (data) of your WE-D320 session. The `Font_R` is used for normal intensity display and the `Font_B` is used for highlight (bold) display. Both fonts **must** exist and **must** be accessible to your window server. For more informations see “Fonts specifications” on page 71.

– Defaults: None

– Mandatory: Yes

26. Font_Dimensions
- Used by X servers only -

– Definition: Here you must place all sizes of `Font_R` and `Font_B` that you want to use in the WE-D320 session. Note that all

the sizes listed have to be available to your server. The sizes can be separated by spaces or commas.

– Defaults: None

– Mandatory: Yes for X windows / No for NEXTSTEP

27. Start_Height

– Definition: Here you have to put the size of `Font_R` and `Font_B` that you want to use at WE-D320 start-up time. For X servers, it must be one of the sizes listed above. For NEXTSTEP, it can be any size.

– Defaults: None

– Mandatory: Yes

28. Font_P

– Definition: This entry defines the font to be used for the buttons in the panels (see “Buttons_Panel (*)” on page 55). The font given here **must** exist and **must** be accessible to your window server. For more informations see “Fonts specifications” on page 71.

– Defaults: None

– Mandatory: Yes

NOTE: The font size chosen here will define the minimum size of the WE-D320 window while the buttons are displayed.

29. Font_Warning

– Definition: If you set this entry to 1 (YES), you will get more detailed messages, if the server has some trouble while loading the fonts you specified. This may be useful if you are suspecting a font loading problem. Possible values are:

1 => Display warnings
0 => Be quiet

– Defaults: 0 (Be quiet)

– Mandatory: No

30. Invert_Bold

– Definition: If you set this entry to 1 (YES), the regular font will be used to display normal intensity characters. The bold font will be used for the low intensity characters. Possible values are:

1 => Invert bold and normal intensity
0 => Normal rendition

– Defaults: 0 (Normal rendition)

– Mandatory: No

WE-D320 CHILD PROGRAM

31. Shell_Program

– Definition: This entry is very important. It will determine which program will be started first within your WE-D320 terminal window. You may start any UNIX process within that window. Most users, however, will start a "shell" like "sh, csh, ksh, etc.". For more informations about this entry, please refer to chapter 4.3 on page 17.

– Defaults: None (You will get a useless WE-D320 window).

– Mandatory: Yes

WE-D320 AUXILIARY FILES

32. Keymap_File

– Definition: The name of the file containing the definitions for your keyboard mapping. You may modify this file using the "KM-D320" utility. Please refer to the "Keymapper tool" on page 43 for a complete description of this utility.

– Defaults: None

– Mandatory: Yes, if you want to map function keys, use international character or macros.

NOTE: We deliver a file called "km.dat" with the emulation. This is a template that you may modify to match your requirements.

33. Macro_File

– Definition: The name of the file containing the definitions for up to 12 keyboard macros. Although you may create or modify this file with a common text editor, we recommend that you use the "Macro definition" panel while running the emulation (see "Macro definitions" on page 36).

– Defaults: None

– Mandatory: Yes, if you want to use macros.

NOTE: We deliver a file called “macro.dat” with the emulation. This is a template that you may modify to match your requirements.

34. Kbd_Layout_File

– Definition: The name of the file containing the image of the terminal’s original keyboard. This image appears in a separate window when you choose the “keyboard layout” entry from the Goodies menu.

– Defaults: None

– Mandatory: Yes, if you want to display the graphical keyboard.

NOTE 1: The same file is also used by the keyboard mapping program.

NOTE 2: This file may not be modified by the user.

35. Color_File

– Definition: The name of the file containing the color definitions for your emulation. It is useful, even if you have no color application, because it allows you to choose custom colors for your window background and foreground.

– Defaults: None

– Mandatory: Yes, if you want custom colors.

NOTE: With the emulation, we include a “color.dat” file as a template. You may use it as is or modify it at will.

36. Buttons_File

– Definition: The name of the file containing the buttons definition for your emulation. Format of this file is as follows:

< Index, Button_title, M_func_name>
< Index, Button_title, M_func_name>
< Index, Button_title, M_func_name>
...

Where:

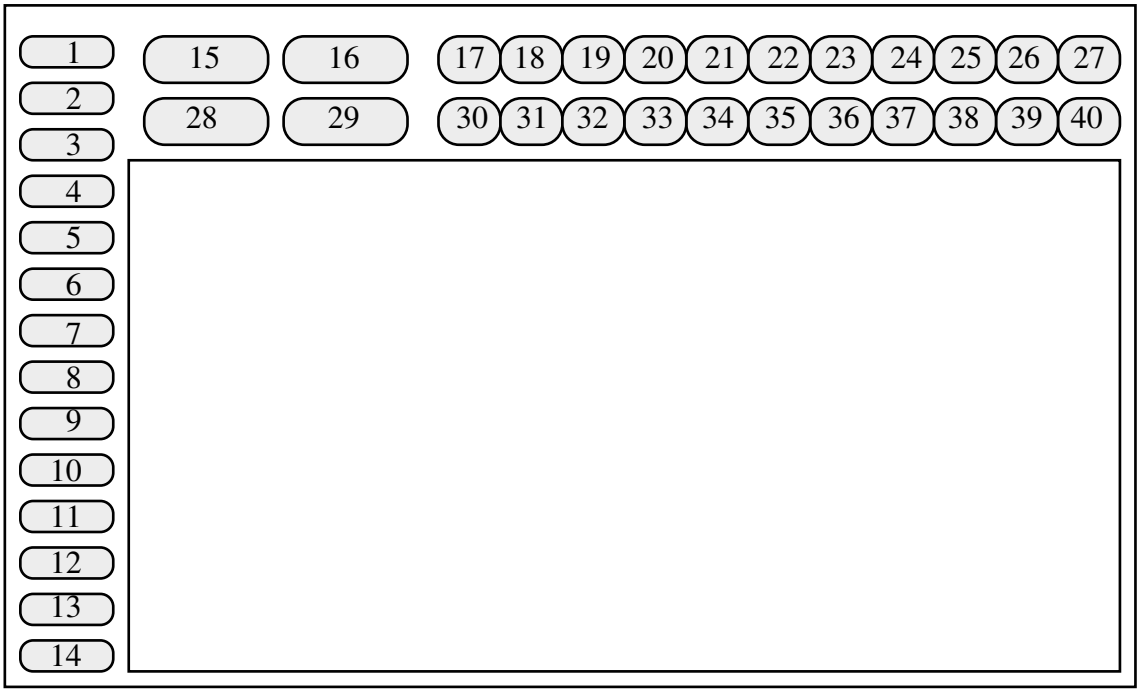
-> “Index” the fix assigned button number as in figure below.

-> “Button_title” the text which will appear in the button.

-> "M_func_name" the emulator function executed when you click this button with your mouse
(see Appendix D on page 81).

– Defaults: None

– Mandatory: Yes, if you want custom buttons.
No, if you are happy with the default text and functions of the buttons.



REMARK: -> If you want to have space chars in the Button_title, you must bracket its definition as follows:

<3, "My Button", M_DO>

Will display My Button in button 3. Clicking this button, will send a DO function code to the attached application.

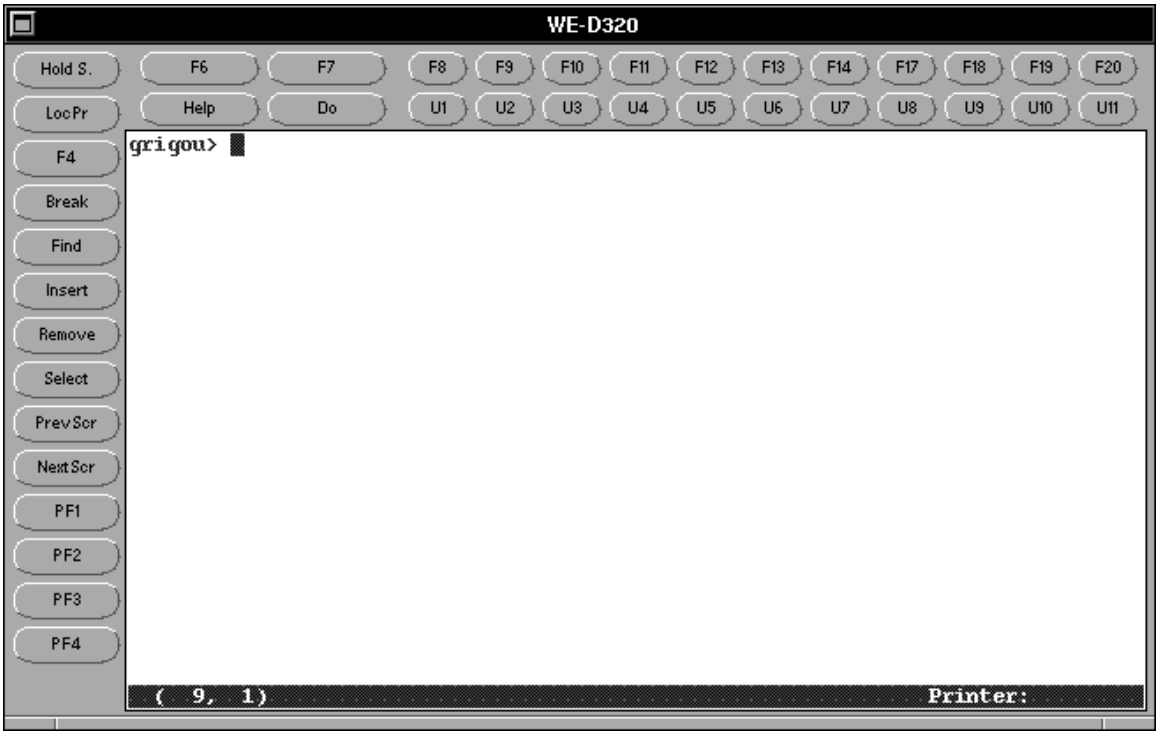
-> On X11 servers, the button width on top-right panel is determined by the length of their title. All other buttons (top-left panel and left panel) have a fixed width.

-> On NeXT, all buttons are of fixed width.

-> Depending on the text length and on the fixed (variable) width of the button, not all text entered may appear in it. Therefore, we encourage you to test your definitions with different window sizes (resize) to help refining them.

-> With the emulation, we deliver a “button.dat” file as a template. You may use it as is or modify it at will.

-> Buttons, for which you give no definitions have defaults as below:



WE-D320 PRINT PARAMETERS

The different print formats are discussed in chapter 5.2.5 on page 22. Please refer to that chapter for detailed informations regarding printing with WE-D320.

37. Disk_Print *

– Definition: The name of the file print data will be directed to when the print destination is a disk file.

– Defaults: /tmp/WE-D320/disk_WE-D320.dat

– Mandatory: No

– Example:

Disk_Print ~/we-d320/MyPrints.$ext

Creates files with sequential numbering in the user's home directory. See discussion in chapter 5.2.5 on page 22 for the meaning of $ext and how it will be replaced by WE-D320.

Disk_Print /tmp/MyPrints

Puts the print data in the file "MyPrints" in the /tmp directory. Since (without $ext) this file will be created at each print occurs, it will only contain the data of the **last print** done.

38. Print_Program *

– Definition: The name of the program print data will be directed to when the print destination is a program.

– Defaults: lpr -r -h $FILE

– Mandatory: No

39. Print_Dest *

– Definition: The default destination for your print data. Possible values are:

Printer
Disk

– Defaults: Printer

– Mandatory: No

40. Print_Status_Line *

– Definition: Will add the status line and a window border on the printout if set. Values are:

0 # No status line
1 # Add status line to printouts

– Defaults: 0 (No status line)

– Mandatory: No

41. Print_Format *

– Definition: Will define in which format printing will occur. Values are:

Text only
Formatted
PostScript

– Defaults: Text only

– Mandatory: No

42. Print_Output_Size *

– Definition: Will define the paper size. This entry is only used if "Print_Format" is set to PostScript. Values are:

Letter
A 3
A 4
A 5

– Defaults: Letter

– Mandatory: No

43. Print_Orientation *

– Definition: Will define the paper orientation. This entry is only used if "Print_Format" is set to PostScript. Values are:

Portrait
Landscape

– Defaults: Portrait

– Mandatory: No

44. Print_Scale *

– Definition: Will define the scaling factor in %. This entry is only used if "Print_Format" is set to PostScript. Values are:

from 30 to 300 (decimal value)

– Defaults: 100

– Mandatory: No

45. Print_Generic_File

– Definition:	Will define the name and path of the PostScript generic file which WE-D320 will use as a template to generate PostScript: **~/we-d320/ps_generic.dat**
– Defaults:	None
– Mandatory:	Yes, if you want to generate PostScript and you don't work with NEXTSTEP.

REMARK: With the emulation, we deliver a file called "ps_generic.dat". This is a template that you probably will use as delivered. Experienced PostScript programmers may modify it to suit their special needs.

WE-D320 PASSWORD

46. Pass_Word

– Definition:	A password provided by workstation ag. This must be typed exactly as provided. A password may contain (depending on your particular licence agreement) informations like: – hostid or hostname – expiration date – maximum run time – host(s) allowed to display
– Defaults:	None
– Mandatory:	Yes

A.2 Fonts specifications

1. All fonts used with "WE-D320" **must** be fixed character width (not proportional) fonts (i.e.: b&h, courier).
2. Both fonts (Font_R and Font_B) must be the same size (width and height).
3. The fonts and sizes you specify must be available to your X or PostScript server.
4. While the Buttons_Panel is displayed, the window can **never** become smaller than the buttons size. So, if you want to resize your window to a smaller size than the buttons, you **must** first disable them with the "Toggle" menu option.

A.3 Fonts recommended for X window systems

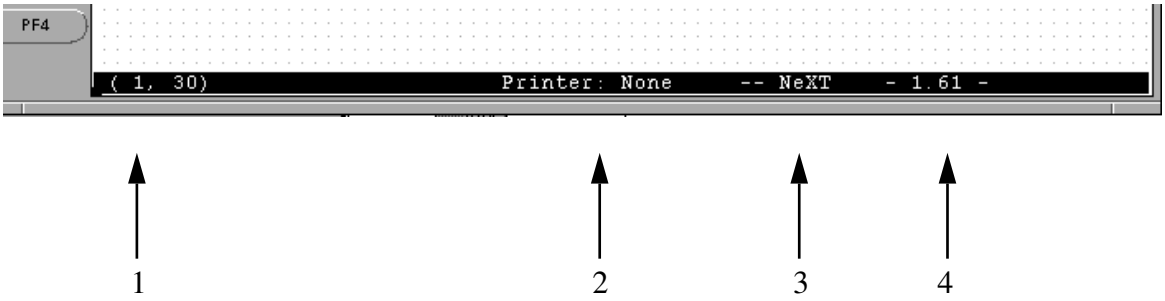
Font_R	-b&h-lucidatypewriter-medium-r-*-*-%d-*-*-*-m-*-iso8859-1
Font_B	-b&h-lucidatypewriter-bold-r-*-*-%d-*-*-*-m-*-iso8859-1
Font_Dimensions	8, 10, 12, 14, 18, 19, 24
Start_Height	12
Font_P	-b&h-lucidatypewriter-medium-r-*-*-%d-*-*-*-m-*-iso8859-1

A.4 Fonts recommended for NEXTSTEP systems

Font_R	Courier
Font_B	Courier-Bold
Font_Dimensions	NOT APPLICABLE
Start_Height	12
Font_P	Helvetica

B Status line, window border line, icon and messages

B.1 The status line contains the following fields



1. Cursor coordinates	Will show cursor location in line / column.
2. Printer status	Reserved to show host controlled printer status. Currently, printing directly from the host application is **not** supported.
3. Target machine	Machine on which WE-D320 is running.
4. Release	WE-D320 software release.

REMARK: The last (mostly the 25th line) of the text window behave as a status line only if its mode is set to “Indicator”. The mode of this line can both be set by:

=> The configuration file (“Status_Line” parameter)
=> The parameter panel (displayed from the Goodies)
=> The host application through escape sequences

B.2 Content of the WE-D320 icon

The WE-D320 have the following aspect:

C Network WE-D320 installation

The following is an illustration how you may install WE-D320 effectively on a network. Although this example is tailored for NeXT machines, the same principles can be applied for any UNIX / X11 based workstation for which we provide emulation software. The goal is to allow **any** user to log on **any** station and use WE-D320 exactly the same way with its own configuration file, colors, macros, etc.

To achieve this goal, the WE-D320 parameters have been split in three parts:

– A part specific to each station containing the station password and the keymap file to use. (Placed in *</etc/we-d320-config/.we-d320-rc>* on each machine.)

– A part providing default values for all remaining parameters. (Placed in *<$InstDir/we-d320.app/Default.D320_config>* on the program server.)

– A part where the user may override any previous defined value with own values. (Placed in *<~/VT320/private.D320_config>*.) "~" stands for the users's home directory.

Thereafter, we will guide you step by step through this installation procedure.

C.1 On the machine used as program server

1. Choose the directory where you already put the applications distributed throughout the network. Thereafter, we will call this directory the *<$InstDir>* directory.

2. Create a *<km-d320.app>* directory in *<$InstDir>* and put the following files therein:

The <$InstDir/we-d320.app> directory

contains the files:

km-d320
keylay.dat

3. Create a *<we-d320.app>* directory in *<$InstDir>* and put the following files therein:

The <$InstDir/we-d320.app> directory

contains the files:

we-d320
keylay.dat
color.dat
macro.dat
US_km.dat
SwissGerman_km.dat
Default.D320_config

REMARK: On a NeXT, the installation procedure will create the above directories for you and put the corresponding files into them. However, you will find a file called "Sample.D320_config" containing all possible parameters for a VT320 emulation. You will be able to use it as a template to create the configuration files described below.

<US_km.dat> and *<SwissGerman_km.dat>* are created with the *<km-d320>* utility as explained in the chapter 6.3 on page 45.

<Default.D320_config> is derived from the delivered *<my.D320_config>* and contains the following entries:

```
# WE-D320 WINDOW PARAMETERS
Term_Name       Vt220       # Name of terminal for "termcap" or "terminfo"
Screen_Rows     24          # WE-D320 window will have 24 rows
Screen_Cols     80          # and 80 columns
Buttons_Panel   1           # The window will be bordered with buttons
Slow_Scroll     1           # Better for your eyes
Exit_Confirm    1           # Avoids unwanted exits. Brings a warning panel
Lock_Setup      0           # The primary configuration file will NOT be locked

# WE-D320 BLINKING OPTIONS
Blink_Cursor    1           # Cursor will blink
Blink_OutFocus  0           # Blinking will occur only while window selected
Inactiv_Time    15          # Any blink activity will stop after 15 seconds

# THOSE MAY BE CHANGED BY THE HOST APPLICATION AS WELL
Cursor_Enable   1           # Yes, we want a cursor displayed
Status_Line     Indicator   # Format of 25th line = status
```

New_Line	0	# Return will send CR alone, not CR/LF
Local_Echo	0	# No, wait for application echoing
Auto_Wrap	1	# Go to next line automatically
Nrc_Mode	0	
Vt_Mode	VT320_8	# Will start in 8 bit control mode
User_Set	DEC	# DEC multinational set
Keypad_Mode	Normal	# Keypad will send 1, 2, 3
Cursor_Keys	Numeric	

THOSE MAY NOT BE CHANGED BY THE HOST APPLICATION

Keyboard_Type	NORTHAMERICAN	# US NRC set will be used
Udk_Lock	0	# Host can change UDK's
Term_Id	VT320	# We will respond as being a VT320
On_Line	1	# Yes, we want to talk to DEC

WE-D320 FONTS

Font_R	Courier	
Font_B	Courier-Bold	
Start_Height	12	
Font_P	Helvetica	
Font_Warning	1	# Will print warnings for font problems
Invert_Bold	0	# Don't permute normal and bold attributes

WE-D320 CHILD PROGRAM

Shell_Program	csh	# Mostly sh, csh, ksh, rlogin, tip, etc.

WE-D320 AUXILIARY FILES

Macro_File	$InstDir/we-d320.app/macro.dat	# Default macros
Kbd_Layout_File	$InstDir/we-d320.app/keylay.dat	# The original DEC keyboard
Color_File	$InstDir/we-d320.app/color.dat	# Default colors

WE-D320 PRINT PARAMETERS

Print_Format	Postscript	# Hardcopy output in PostScript format
Print_Status_Line	1	# Add status line and frame to output

The following useful only if "Print_Format" IS NOT "PostScript"

Disk_Print	~/we-d320.$ext	# File to print on
Print_Program	lpr -r -h $File	# Print program to fork
Print_Dest	Disk	# Where print data will go

The following useful only if "Print_Format" IS "PostScript"

Print_Output_Size	A4	# Generate PostScript for A 4 paper
Print_Orientation	Landscape	# Mostly the best solution
Print_Scale	100	# A 1/1 representation

C.2 On each machine in the </etc> directory

1. Create a *</etc/we-d320-config>* directory.
2. Create a *</etc/we-d320-config/.we-d320-rc>* file.

The </etc/we-d320-config/.we-d320-rc> file

contains:

Pass_Word	MachineSpecificPasswordHere
Keymap_File	$InstDir/we-d320.app/US_km.dat (or other map)

As shown above, we recommend that you put machine specific informations in the above configuration file. *<Pass_Word>* as well as *<Keymap_File>* (containing keymapping informations for a certain kind of keyboard) are typically machine relative informations.

C.3 In each users's home directory

1. Create a *<VT320>* directory owned by the user (VT320 is an example).
2. Create a *<.we-d320-rc>* link to *<$InstDir/we-d320.app/Default.D320_config>*. This **link** should be owned by *<root>* to avoid any modification by the user.

REMARK: The link may be created with the command:

ln -s $InstDir/we-d320.app/Default.D320_config .we-d320-rc

The <~> directory (~ stand for user's home directory)

contains the link to the default configuration file:

.we-d320-rc -> $InstDir/we-d320.app/Default.D320_config

The <~/VT320> directory

contains the files:

private.D320_config	(you may create several such files)
private.color.dat	(the user's custom colors)
private.macro.dat	(the user's custom macros)

File <private.D320_config> may contain the following entries:

```
# WE-D320 WINDOW PARAMETERS
Screen_Rows     24              # WE-D320 window will have 24 rows
Screen_Cols     80              # and 80 columns

# WE-D320 FONTS
Start_Height    12

# WE-D320 CHILD PROGRAM
Shell_Program   csh             # Mostly sh, csh, ksh, rlogin, tip, etc.

# WE-D320 AUXILIARY FILES
Macro_File      ~/VT320/private.macro.dat   # Your favorite macros
Color_File      ~/VT320/private.color.dat   # Your favorite colors

# WE-D320 PRINT PARAMETERS
Disk_Print      ~/we-d320.$ext  # File to print on
Print_Dest      Disk            # Print destination is disk
```

C.4 How to start WE-D320 (valid for NeXT only)

For starting WE-D320, you may click either on any *<xxx.D320_config>* file or on *<we-d320.app>* itself.

If you click on the <we-d320.app> application icon in a browser or on the dock:

WE-D320 will start reading the following <config> files in that order:

=> */etc/we-d320-config/.we-d320-rc*

=> *~/.we-d320-rc* (which is linked to "Default.D320_config")

This means that WE-D320 will start with the default configuration.

If you click on any <private.D320_config> file (i.e.: mypri1.D320_config):

WE-D320 will start reading the following <config> files in that order:

=> */etc/we-d320-config/.we-d320-rc*

=> *~/.we-d320-rc* (which is linked to "Default.D320_config")

=> *mypri1.D320_config*

This means that WE-D320 will start with one of your private configuration.

NOTE: The NeXT workspace looks for applications in some predefined areas like

The Dock
~/Apps
/LocalApps
/NeXT Apps
...

in an order defined by NeXT (see the NeXT "Users's Reference Manual").

This means that <*we-d320*.app> (as any other application) **must** reside in one of these special areas to be recognized as application by the Workspace Manager. If this is not the case, the <xxx.D320_config> files will appear as text files instead of having the WE-D320 icon and you will not be able to start WE-D320 by clicking on them. (Editor will be called instead.) To achieve this, you may:

=> Put <*we-d320*.app> on the dock (the easiest way).

=> Mount the directory exported by the application server in *</LocalApps>* instead of *</Net/ServerName/Directory>*. This will allow the workspace to recognize any <.app> contained therein.

D "M_func_names" list

A list of the names that can be used in the optional "Buttons_File":

M_HOLD_SCREEN	Will send Ctrl/q, Ctrl/s to the host (toggle function)
M_PRINT_SCREEN	Screen hardcopy (with current format)
M_F3	Same as DEC VT320 F3
M_F4	Same as DEC VT320 F4
M_BREAK	Currently no function
M_F6	Same as DEC VT320 F6
M_F7	Same as DEC VT320 F7
M_F8	Same as DEC VT320 F8
M_F9	Same as DEC VT320 F9
M_F10	Same as DEC VT320 F10
M_F11	Same as DEC VT320 F11
M_F12	Same as DEC VT320 F12
M_F13	Same as DEC VT320 F13
M_F14	Same as DEC VT320 F14
M_HELP	Same as DEC VT320 Help key
M_DO	Same as DEC VT320 Do key
M_F17	Same as DEC VT320 F17
M_F18	Same as DEC VT320 F18
M_F19	Same as DEC VT320 F19
M_F20	Same as DEC VT320 F20
M_DEL	Delete key
M_TAB	Tab key
M_RETURN	Return key
M_SPACE_BAR	Space key
M_FIND	Same as DEC VT320 Find key
M_INSHERE	Same as DEC VT320 Insert key
M_REMOVE	Same as DEC VT320 Remove key
M_SELECT	Same as DEC VT320 Select key
M_PREVSCR	Same as DEC VT320 PrevScr key
M_NEXTSCR	Same as DEC VT320 NextScr key
M_UP	Same as Up arrow key
M_LEFT	Same as Left arrow key
M_RIGHT	Same as Right arrow key
M_DOWN	Same as Down arrow key

M_PF1	Same as DEC VT320 PF1
M_PF2	Same as DEC VT320 PF2
M_PF3	Same as DEC VT320 PF3
M_PF4	Same as DEC VT320 PF4
M_PAD0	Same as DEC VT320 keypad 0
M_PAD1	Same as DEC VT320 keypad 1
M_PAD2	Same as DEC VT320 keypad 2
M_PAD3	Same as DEC VT320 keypad 3
M_PAD4	Same as DEC VT320 keypad 4
M_PAD5	Same as DEC VT320 keypad 5
M_PAD6	Same as DEC VT320 keypad 6
M_PAD7	Same as DEC VT320 keypad 7
M_PAD8	Same as DEC VT320 keypad 8
M_PAD9	Same as DEC VT320 keypad 9
M_PADPOINT	Same as DEC VT320 keypad point
M_PADMINUS	Same as DEC VT320 keypad minus
M_PADVIRGULE	Same as DEC VT320 keypad comma
M_ENTER	Same as DEC VT320 Enter key
M_UDK1	Same as DEC VT320 user definable key 1
M_UDK2	Same as DEC VT320 user definable key 2
M_UDK3	Same as DEC VT320 user definable key 3
M_UDK4	Same as DEC VT320 user definable key 4
M_UDK5	Same as DEC VT320 user definable key 5
M_UDK6	Same as DEC VT320 user definable key 6
M_UDK7	Same as DEC VT320 user definable key 7
M_UDK8	Same as DEC VT320 user definable key 8
M_UDK9	Same as DEC VT320 user definable key 9
M_UDK10	Same as DEC VT320 user definable key 10
M_UDK11	Same as DEC VT320 user definable key 11
M_UDK12	Same as DEC VT320 user definable key 12
M_UDK13	Same as DEC VT320 user definable key 13
M_UDK14	Same as DEC VT320 user definable key 14
M_UDK15	Same as DEC VT320 user definable key 15
M_UDK16	Same as DEC VT320 user definable key 16
M_UDK17	Same as DEC VT320 user definable key 17
M_UDK18	Same as DEC VT320 user definable key 18
M_UDK19	Same as DEC VT320 user definable key 19
M_UDK20	Same as DEC VT320 user definable key 20
M_MACRO1EXEC	Will execute your custom macro No 1 (see macro panel)
M_MACRO2EXEC	Will execute your custom macro No 2
M_MACRO3EXEC	Will execute your custom macro No 3
M_MACRO4EXEC	Will execute your custom macro No 4
M_MACRO5EXEC	Will execute your custom macro No 5
M_MACRO6EXEC	Will execute your custom macro No 6
M_MACRO7EXEC	Will execute your custom macro No 7
M_MACRO8EXEC	Will execute your custom macro No 8

M_MACRO9EXEC	Will execute your custom macro No 9
M_MACRO10EXEC	Will execute your custom macro No 10
M_MACRO11EXEC	Will execute your custom macro No 11
M_MACRO12EXEC	Will execute your custom macro No 12
M_COPY	Will copy the current selection to the pasteboard
M_PASTE	Will copy the pasteboard at the cursor location
M_NUL	Will send a hex 00 to the host
M_CTRL_A	Will send a hex 01 (as if you type Ctrl/a on a VT320)
M_CTRL_B	Will send a hex 02 (as if you type Ctrl/b on a VT320)
M_CTRL_C	Will send a hex 03 (as if you type Ctrl/c on a VT320)
M_CTRL_D	Will send a hex 04 (as if you type Ctrl/d on a VT320)
M_CTRL_E	Will send a hex 05 (as if you type Ctrl/e on a VT320)
M_CTRL_F	Will send a hex 06 (as if you type Ctrl/f on a VT320)
M_CTRL_G	Will send a hex 07 (as if you type Ctrl/g on a VT320)
M_CTRL_H	Will send a hex 08 (as if you type Ctrl/h on a VT320)
M_CTRL_I	Will send a hex 09 (as if you type Ctrl/i on a VT320)
M_CTRL_J	Will send a hex 0a (as if you type Ctrl/j on a VT320)
M_CTRL_K	Will send a hex 0b (as if you type Ctrl/k on a VT320)
M_CTRL_L	Will send a hex 0c (as if you type Ctrl/l on a VT320)
M_CTRL_M	Will send a hex 0d (as if you type Ctrl/m on a VT320)
M_CTRL_N	Will send a hex 0e (as if you type Ctrl/n on a VT320)
M_CTRL_O	Will send a hex 0f (as if you type Ctrl/o on a VT320)
M_CTRL_P	Will send a hex 10 (as if you type Ctrl/p on a VT320)
M_CTRL_Q	Will send a hex 11 (as if you type Ctrl/q on a VT320)
M_CTRL_R	Will send a hex 12 (as if you type Ctrl/r on a VT320)
M_CTRL_S	Will send a hex 13 (as if you type Ctrl/s on a VT320)
M_CTRL_T	Will send a hex 14 (as if you type Ctrl/t on a VT320)
M_CTRL_U	Will send a hex 15 (as if you type Ctrl/u on a VT320)

M_CTRL_V	Will send a hex 16 (as if you type Ctrl/v on a VT320)
M_CTRL_W	Will send a hex 17 (as if you type Ctrl/w on a VT320)
M_CTRL_X	Will send a hex 18 (as if you type Ctrl/x on a VT320)
M_CTRL_Y	Will send a hex 19 (as if you type Ctrl/y on a VT320)
M_CTRL_Z	Will send a hex 1a (as if you type Ctrl/z on a VT320)
M_ESC	Will send a hex 1b (as if you type Esc on a VT320)
M_FS	Will send a hex 1c to the host
M_GS	Will send a hex 1d to the host
M_RS	Will send a hex 1e to the host
M_US	Will send a hex 1f to the host

Index

A

B

C

D

E

F

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W