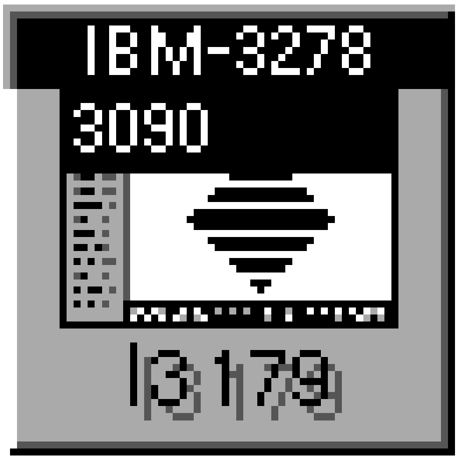
WE-I3179

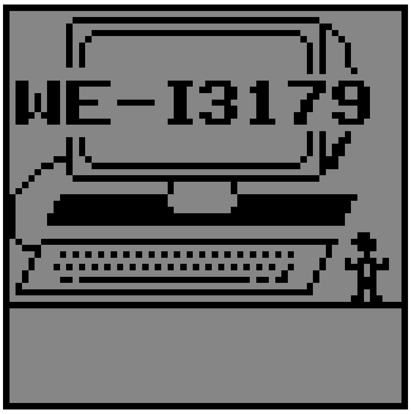
Installation and User Reference Manual

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.

Table of Contents

1 Preface

1.1 Purpose and audience

This manual allows UNIX workstations to interact with IBM™ host applications designed for IBM™ terminals running in 3270 mode. This is a user's guide, it is aimed at end users who are UNIX users and are familiar with using the IBM™ display terminal and printer. The typical audience is also familiar with IBM™ host login and the host applications accessible by the IBM™ display terminal product.

1.2 Summary of content

Chapter 2: "Overview"
presents an overview of the WE-I3179 terminal emulation package. Included are possible configurations of the software package, supported features, and system software requirements.

Chapter 3: "Installing WE-I3179"
describes the tape contents and installation procedure for WE-I3179.

Chapter 4: "Configuring WE-I3179"
describes the general considerations for configuring workstations and terminals.

Chapter 5: "Considerations about locking config file(s)"
describes how to use locking mechanisms for the config files.

Chapter 6: "Using WE-I3179 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 7: "Keymapper tool"
explains how to use the keyboard mapping utility KM-I3179 delivered with your emulation.

Appendix A: "WE-I3179 parameter file format"
lists an example WE-I3179 parameter file and explains each entry.

Appendix B: "About colors and graphics"
explains how to use colors and graphics with your WE-I3179 emulation.

Appendix C: "TELNET example"
shows the relations between the WE-I3179 components in a TCP/IP TELNET environment.

Appendix D: "Status line, window border line, icon and messages"
provides description of the WE-I3179 messages areas and their respective content.

Appendix E: “M_func_names” list
provides a list (and brief description) of all WE-I3179 function names which can be mapped to the buttons placed around the window (see “Buttons_File” on page 67).

1.3 Conventions used

Throughout this manual we use the following conventions:

The *host system* is the IBM™ mainframe computer to which the WE-I3179 device emulator communicates through the WE-COMD server.

The *WE-COMD process* is the server used to emulate the IBM™ communication protocol.

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:

IBM™ publications

1.	GA-23-0059-05	3270 Information Display System, Data Stream Programmer's Reference
2.	GA-27-3831-04	3174 Establishment Controller, Character Set Reference
3.	GA-23-0218-08	3174 Establishment Controller, Functional Description
4.	GA-18-226-2261-0	IBM 3179G Color Graphics Display Station Description

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-I3179 product line centers around three products that allow workstations to communicate with IBM™ mainframes over TCP/IP Ethernet links (TELNET connections), SDLC or X.25 public or private networks:

#	WE-I3179c	Terminal emulation without graphics
#	WE-I3179g	Terminal emulation with graphics
#	WE-COMD	Communication protocol server

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

2.2 Configurations

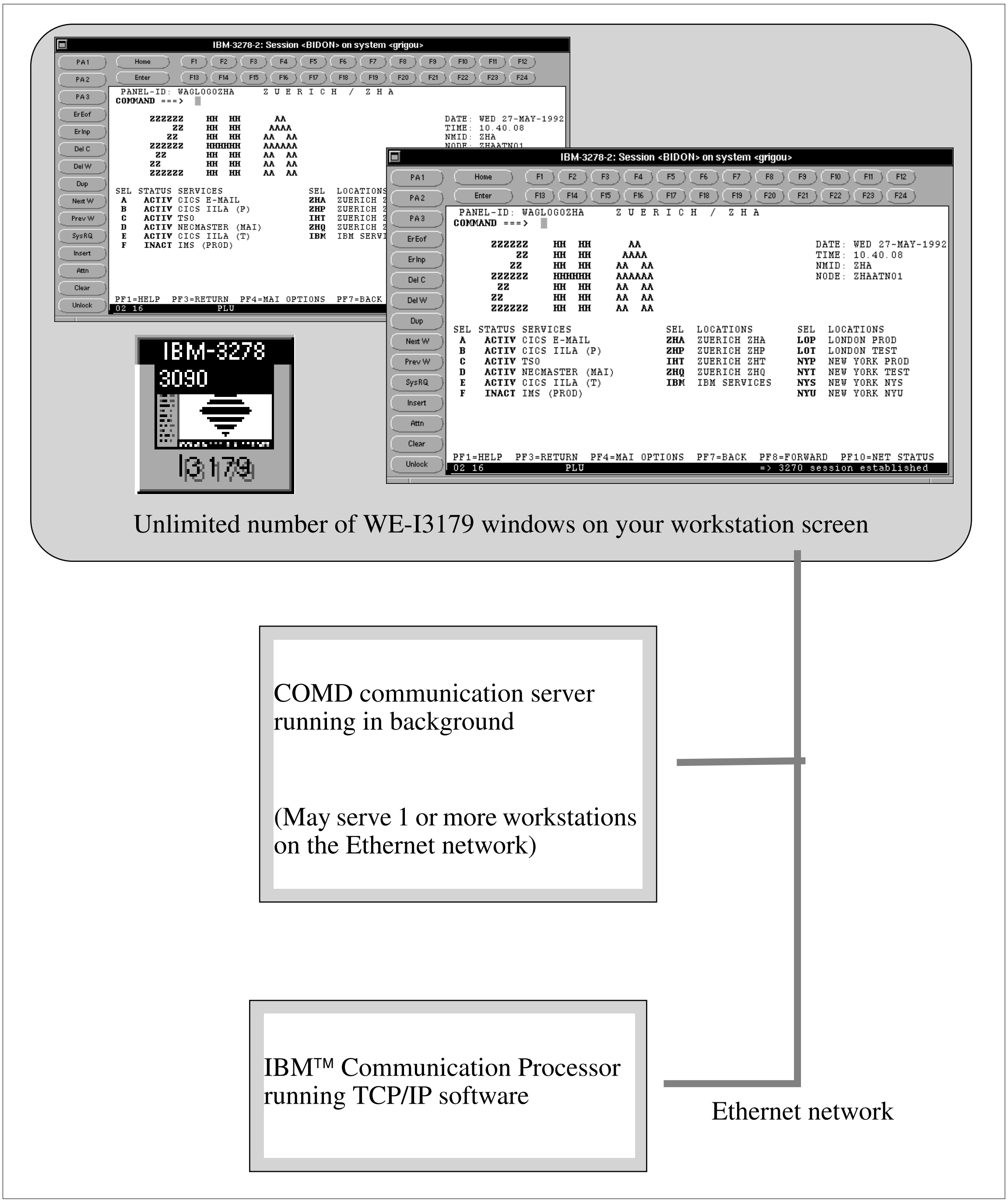
The products are software packages configured to run directly on workstation processors. The software for each product has a gateway component and a workstation component. When a WE-COMD gateway component is installed on a UNIX processor, that processor may become a protocol server for the entire network. The gateway may be shared by any number of workstations, each running any number of WE-I3179 terminal windows.

Each of these WE-I3179 products isolates the communication protocol support to their respective gateways, thereby buffering the workstations from the complexities and additional overhead of TCP/IP TELNET, SDLC or X.25 protocols.

The workstation talks to the gateway using TCP/IP, which provides flexible and high-performance distribution of the WE-I3179 terminal sessions.

2.2.1 Typical examples

The following figure highlights the software components of the WE-I3179 products.



IBM-3278-2: Session <BIDON> on system <grigou>
PA1
PA2
PA3
ErEof
ErInp
Del C
Del W
Dup
Next W
Prev W
SysRQ
Insert
Attn
Clear
Unlock
Home
Enter
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24
IBM-3278
3090
I3179

```
 PANEL-ID: WAGLOGOZHA     Z U E R I C H  /  Z H A
COMMAND ===>

      ZZZZZZ    HH  HH      AA                                   DATE: WED 27-MAY-1992
          ZZ    HH  HH     AAAA                                  TIME: 10.40.08
         ZZ     HH  HH    AA  AA                                 NMID: ZHA
      ZZZZZZ    HHHHHH    AAAAAA                                 NODE: ZHAATN01
       ZZ       HH  HH    AA  AA
      ZZ        HH  HH    AA  AA
      ZZZZZZ    HH  HH    AA  AA

SEL STATUS SERVICES                SEL  LOCATIONS       SEL  LOCATIONS
 A   ACTIV CICS E-MAIL             ZHA  ZUERICH ZHA     LOP  LONDON PROD
 B   ACTIV CICS IILA (P)           ZHP  ZUERICH ZHP     LOT  LONDON TEST
 C   ACTIV TSO                     IHT  ZUERICH ZHT     NYP  NEW YORK PROD
 D   ACTIV NECMASTER (MAI)         ZHQ  ZUERICH ZHQ     NYT  NEW YORK TEST
 E   ACTIV CICS IILA (T)           IBM  IBM SERVICES    NYS  NEW YORK NYS
 F   INACT IMS (PROD)                                   NYU  NEW YORK NYU



PF1=HELP  PF3=RETURN  PF4=MAI OPTIONS  PF7=BACK  PF8=FORWARD  PF10=NET STATUS
02 16           PLU                                   => 3270 session established
```

Unlimited number of WE-I3179 windows on your workstation screen

COMD communication server running in background

(May serve 1 or more workstations on the Ethernet network)

IBM™ Communication Processor running TCP/IP software

Ethernet network

WE-I3179 emulation components for Ethernet (TELNET) connection

2.3 Feature summary

Workstation WE-I3179 terminal emulation offers support for applications written for the 3278 models 2-5, 3279 models 2-5 and 3179G (WE-I3179g color graphic only) IBM terminal families.

2.3.1 WE-I3179c basic datastream operations

All operations accepted by the 3179 color terminal like:

Cursor positioning operations
Structured field operations
Extended highlighting operations
All attribute combinations (including blink)
Read operations
Peripheral control, etc.

2.3.2 WE-I3179g datastream operations

All operations accepted by the WE-I3179c.

Graphic orders in the 3179G format.

2.3.3 WE-I3179 keyboard support

All functions of the WE-I3179 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).

2.3.4 Limitations against IBM™ 3179

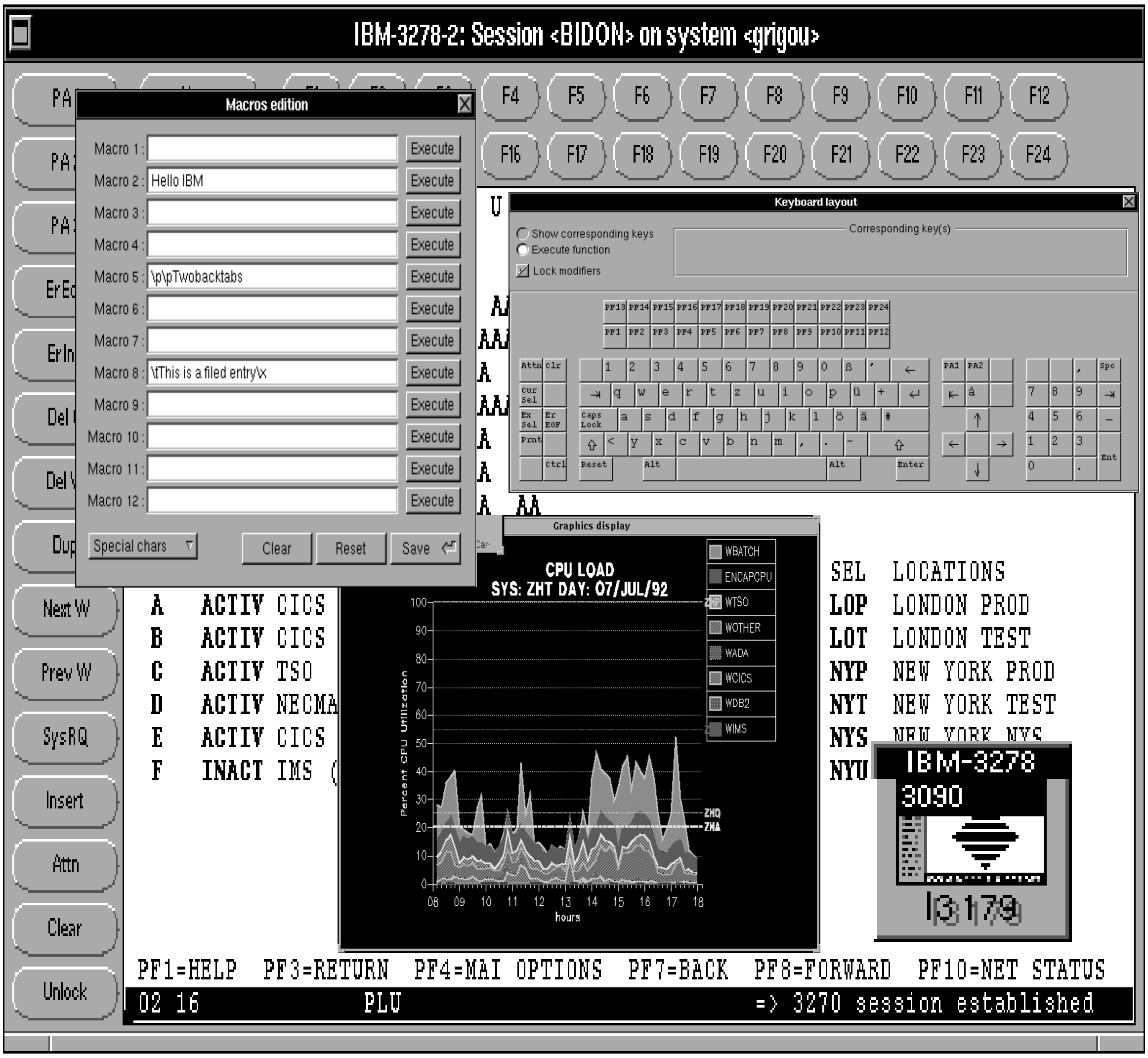
Peripheral operations are limited to the printer. Selector light pen or magnetic slot reader are not supported.

Programmed symbols are not supported.

2.3.5 Additional features not provided by IBM™ terminals

- Unlimited number of simultaneous WE-I3179 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-I3179 windows and other windows.
- Context sensitive help for all keys and parameters.
- Keyboard mapping and up to 12 macro functions can be defined separately for each WE-I3179 window.
- Cursor positioning either by key depression (i.e.: tabs) or by mouse clicking anywhere in the WE-I3179 window.

- A graphical representation of an original IBM 3270 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.



IBM-3278-2: Session <BIDON> on system <grigou>

Macros edition

Macro 1 :
Macro 2 : Hello IBM
Macro 3 :
Macro 4 :
Macro 5 : \p\pTwobacktabs
Macro 6 :
Macro 7 :
Macro 8 : \tThis is a filed entry\x
Macro 9 :
Macro 10 :
Macro 11 :
Macro 12 :

Keyboard layout

○ Show corresponding keys
○ Execute function
☑ Lock modifiers

```
A   ACTIV CICS                              SEL  LOCATIONS
B   ACTIV CICS                              LOP  LONDON PROD
C   ACTIV TSO                               LOT  LONDON TEST
D   ACTIV NECMA                             NYP  NEW YORK PROD
E   ACTIV CICS                              NYT  NEW YORK TEST
F   INACT IMS (                             NYS  NEW YORK NYS
                                            NYU
PF1=HELP  PF3=RETURN  PF4=MAI OPTIONS  PF7=BACK  PF8=FORWARD  PF10=NET STATUS
02 16            PLU                         => 3270 session established
```

3 Installing WE-I3179

3.1 TCP/IP TELNET host connection using the “WE-COMD” communication server

The installation procedure consists of 4 consecutive steps as follows:

a) Extracting the product from the distribution media.

b) Adding entries to your <etc/hosts> and <etc/services> files to declare the different INTERNET names and TCP ports used for the communication between the WE-I3179 components and the IBM™ host (communication controller running TCP/IP).

c) Creating configuration files which will be used by the WE-I3179 terminal emulation to configure itself and address the proper IBM™ host for the sessions you want to establish.

d) Installing the <WE-I3179.info> file for On-line help.

3.2 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-I3179
we-i3179	The terminal emulator
we-comd	The communication server
km-i3179	The keymapping tool
Sample.I3179g_config	A sample config file for I3179 (graphic)
Sample.I3179c_config	A sample config file for I3179 (text only)
keylay.dat	An ASCII image of the IBM terminal keyboard
color.dat	A sample color file
macro.dat	A sample macro file
button.dat	A sample button file
WE-I3179.info	Not provided with Release 1.63x

If you purchased WE-I3179 for an X11 server machine, the following supplemental files will be provided:

ps_generic.dat	Used to generate PostScript hardcopies
cour.pfa	Scalable font used for graphics (standard)
courb.pfa	Scalable font used for graphics (bold)
couri.pfa	Scalable font used for graphics (italic)

3.3 Adding <hosts> and <services> entries for WE-I3179

According to your own network configuration you will have to add correct entries into /etc/hosts and /etc/services (may be on your YP/NIS or Netinfo server) to allow the WE-COMD and the WE-I3179 programs to exchange informations between each other and with the IBM™ communication controller.

Appendix C shows an example for the relationships between the emulation components for a TCP/IP TELNET communication path. Thereafter, you will find some more indications about the important points:

- The <WE-COMD> and <WE-I3179> components need a TCP port to communicate together. You can assign any free port number and service name for that purpose. For example, add the following entry into </etc/services>, YP/NIS, Netinfo, etc.:

```
        we-comd_serv        1031/tcp
```

- The <WE-COMD> and the <IBM communication controller> need a TCP port to communicate together. For a TCP/IP TELNET connection, we will use the <telnet> entry which already exists in all TCP/IP implementations. The following entry should therefore already be present in </etc/services>, YP/NIS, Netinfo, etc.:

```
        telnet              23/tcp
```

- Finally, you must ensure that the workstations on which the WE-I3179 emulation components will be running as well as the IBM communication controller(s) you will connect to are declared in the </etc/hosts> files.

3.4 Creating configuration file(s) for WE-I3179

For a complete description of all possible entries in a parameter file, please refer to Appendix A on page 57. Thereafter, we will discuss where and how WE-I3179 searches for parameter file(s) and some general rules valid for all entries in these files.

3.4.1 Format of an entry in a parameter file

All entries have the format:

ParameterName	*Value*	*Comment*
i.e.: Comd_Host	cracker	# my workstation

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

```
Comd_Host
```

with no *Value* would set `Comd_Host` to **nothing**. This would, of course, not work ...

3.4.2 Up to four parameter files are possible

When started, WE-I3179 looks sequentially for 4 parameter files (all have the format specified in Appendix A on page 57).

– First, it looks for a file with the name <**.we-i3179-rc**> in the <**/etc/we-i3179-config**> directory. If it is found, the parameters contained are extracted and validated.

– Secondly, it looks for a file with the name <**.we-i3179-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 4 parameter files. This is provided to give you more flexibility to divide between common parameters for all WE-I3179 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.

3.5 Installing the <WE-I3179.info> file for On-line help

The <WE-I3179.info> file provides help information for WE-I3179. You may use it on any X window system. Here is how WE-I3179 searches for this file:

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

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

  – Set HELPPATH and copy the <WE-I3179.info> file there.
  – Copy the <WE-I3179.info> file into </usr/lib/help> (default).

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

4 Configuring WE-I3179

After you have created the entries in the UNIX "/etc/services" and "/etc/hosts" files (or updated YP/NIS, Netinfo, etc.) and edited your config file(s), it's now time to try your first connection to the IBM host. In fact, the IBM device we will connect to is not the host itself, but it's "Communication Controller" component which is connected to the Ethernet network.

WE-I3179 will be started in 5 phases as outlined below

1. *Start the WE-COMD component (communication server)*
   This is done once on each workstation (sometimes once for the whole network). For that reason, starting of the WE-COMD server is mostly done automatically in the <rc.local> shell script.

2. *Start the "WE-I3179" 3270 terminal emulator*
   Is done upon request of the user. You can start as many WE-I3179 emulators as you want on a single workstation, as long as you have a valid license that covers that many concurrent windows.

3. *1st (automatic) connection to the host*
   Occurs only if all parameters necessary for a connection were found in the config file(s).

4. *Use the <reconnection> Main menu entry*
   Can occur as often as you want. Since choosing this entry will disconnect you from the IBM and make a new connection, a panel is displayed and give you a last chance to cancel this action.

5. *Start communication controller simulator*
   This is a separate program deliverable upon request.

4.1 Starting the "WE-COMD" communication server component

The syntax for starting the WE-COMD component is as follows:

we-comd ServiceName [-sm] [-lc] [-t type] [-lp LogFilePath]

1. *<ServiceName>*
   Must be the first parameter. This service will be used for communication between <we-<comd> and <we-i3179>. His name must be the same as the <Comd_Service> entry in the WE-I3179 parameter file. It must also be added to the </etc/services> file (or YP, Netinfo).

2. *<-sm>*
   Optional (means short messages and is optional) will restrict "WE-COMD" to print less detailed information messages.

3. *<-lc>*
   Optional (means log to console) WE-COMD write all the messages to the console. Normally, WE-COMD creates a file (see “<-lp LogFilePath>” on page 20).

   </tmp/we-comd.PID> where PID is the process ID

   at start-up time. If errors occur, messages will be written into that file. If you start WE-COMD with the *[lc]* option, no error logging file will be opened and the messages will be sent to the console instead.

REMARK: Depending on your particular system, writing messages on the console may destroy the appearance of your graphical environment. Mostly, this can be repaired with a <refresh> function of your window manager.

4. *<-t type>*
   Where type is the following:

   tn3270 (used for TCP/IP TELNET connection)

5. *<-lp LogFilePath>*

   Where LogFilePath may be any existing directory where you may create new files:

   Example:

   ```
   machine% we-comd we_serv -t tn3270 -lp /Log
   ```

   When WE-COMD starts, it will print a message like:

   ```
   <we-comd 3.24 pid:4279> at port <1031> in mode <tn3270>

   <we-comd 3.24> error logging will be on </Log/we-comd.4279>
   ```

   This means that WE-COMD is running and is waiting for WE-I3179 connections at TCP/IP port 1031 (the port you have assigned for “Comd_Service”). Afterwards, WE-COMD does its job in the background and you will get **no** messages unless any special condition occurs. Error logging will occur on file </Log/we-comd.4279>.

   If the <ServiceName> obligatory parameter is missing or wrong you will get either of the following messages:

   For example, if you type

   we-comd

   you get the following message:

   ```
   we-comd: Correct syntax is: we-comd ServiceName
              [-sm] [-lc] [-t type] [-lp LogFilePath]
   ```

If you type

```
          we-comd far_west
```

and service <far_west> doesn't exist, you will get the following message:

```
we-comd: far_west not a server
```

4.1.1 Hints for starting the "WE-COMD" server from the "rc" file

Under UNIX, most daemons are started at the end of the boot procedure. WE-COMD is no exception and may also be started there. All parameters described above remain valid. Depending on your particular system, you may add the following line into </etc/rc>, </etc/rc.local>, etc.

```
          we-comd comd-serv -sm -t tp0 >> /tmp/we-comd.startup 2>&1
```

This would start <we-comd> in <tp0> mode and redirect any messages occurring at start-up time to the file </tmp/we-comd.startup>. Redirecting start-up messages to a file may be useful for debugging purposes.

4.2 Starting the "WE-I3179" terminal emulator component(s)

The syntax for starting the "WE-I3179" component is as follows:

we-i3179 [*FileName*][**-sf** *FileName*]

1. <*'FileName'*>   File name of configuration file for WE-I3179.
2. <-sf *'FileName'*>   Special configuration file for WE-I3179.

All the configuration files have the same format (see "Creating configuration file(s) for WE-I3179" on page 16).

Example:

```
          machine% we-i3179 /usr/we-i3179/john/myconfig
                      -sf /usr/we-i3179/all/grpconfig
```

Then, "WE-I3179" will try to read (in sequence):

– A file named <.we-i3179-rc> in the </etc/we-i3179-config> directory.

– A file named <.we-i3179-rc> in your home directory.

– A file named <grpconfig> in the </usr/we-i3179/all> directory.

– A file named <myconfig> in the </usr/we-i3179/john> directory.

NOTE 1: All the 4 possible parameter files have the same format which is detailed under Appendix A on page 57.

NOTE 2: If any mandatory parameter is missing, you will get an error message or a panel message in the form:

```
Configuration problem
Missing parameters in configuration file:
```

NOTE 3: If all mandatory parameters were present and valid you will get a WE-I3179 window on your workstation screen.

NOTE 4: A first connection to the host will be attempted.

NOTE 5: If any parameter was invalid or the necessary fonts were not found or any error occurred at start time, you will **not** get any WE-I3179 window but one of the error messages.

4.2.1 Error messages while starting the “WE-I3179” terminal emulator

The following are fatal errors which can occur when starting the WE-I3179 terminal emulator component. When they occur, **no** WE-I3179 window will be created and you will have to correct the error(s) before proceeding.

a) If obligatory parameter(s) is (are) missing, you will get a message like:

```
WE-I3179:Missing parameter(s):
- Comd_Service
- Mainframe_Service
```

Detailing all missing (obligatory) parameters.

Action => Add the corresponding parameters in a config file.

b) If the <Comd_Host> from the configuration file(s) can’t be found, you will get the following message:

```
WE-I3179:init_socket:<HostName>:unknown host
```

Where: <HostName> is the name found in the config file(s).

Action => Correct the <Comd_Host> entry in the config file so it matches a valid entry in </etc/hosts> (YP or Netinfo).

c) If the <Comd_Service> from the configuration file(s) can’t be found, you will get the following message:

```
WE-I3179:init_socket:<ServiceName>:not a server
```

Where: <ServiceName> is the name found in the config file(s).

Action => Correct the <Comd_Service> entry in the config file so it matches a valid entry in </etc/services> (YP or Netinfo).

d) If the protocol server <WE-COMD> is not running at all or a connection to it is impossible, you will get the message:

```
WE-I3179:init_socket:connect: Connection refused
```

Action => Start the <WE-COMD> component on the <Comd_Host> machine under the <Comd_Service> declared in the config file(s). Consult previous chapters.

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

```
WE-I3179: Font <FontName> NOT found
```

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

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

4.2.2 Note for NeXT users

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

– The application (we-i3179) is installed in a directory known by the Workspace Manager like:

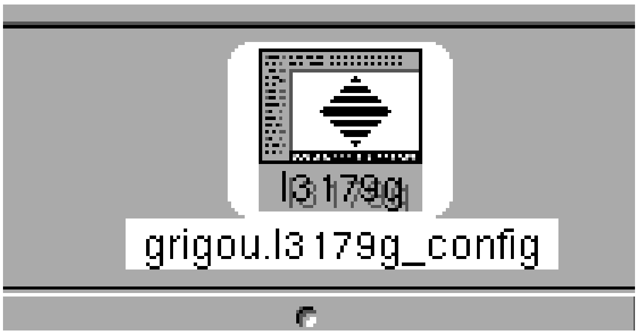
/Apps
~/Apps
/LocalApps
...

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

.I3179g_config (if you purchased a WE-I3179g <3179G>) emulation

.I3179c_config (if you purchased a WE-I3179c <3179>) emulation

If both conditions are met, the <xxx.I3179c_config> file will appear in your browser with the same icon as the <we-i3179> 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-I3179> will still try to read

– A file named <.we-i3179-rc> in the </etc/we-i3179-config> directory
– A file named <.we-i3179-rc> in your home directory

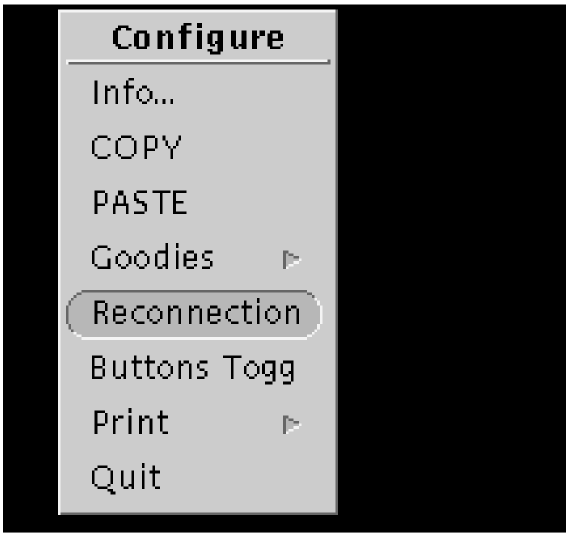
to get configuration information. The config file you click on will be read last and parameters contained herein may overwrite previous defined values.

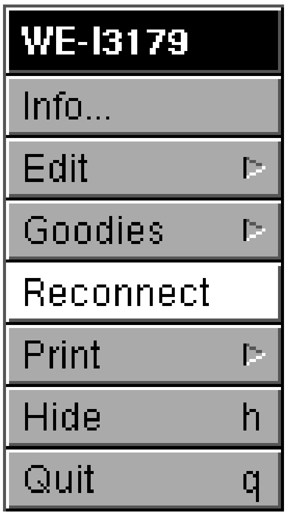
4.3 Establishing a WE-I3179 to host connection

A connection to the host is established:

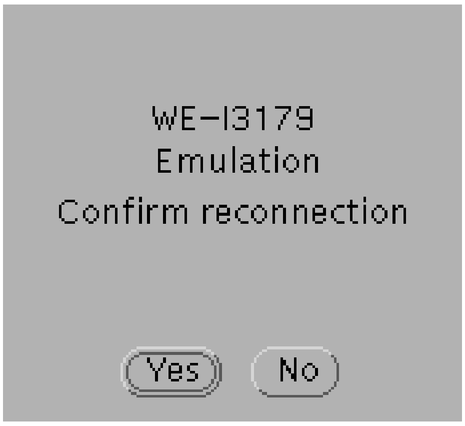
a) Automatically at WE-I3179 start-up.

b) At any time by choosing the <Reconnect> entry in the WE-I3179 Main menu.

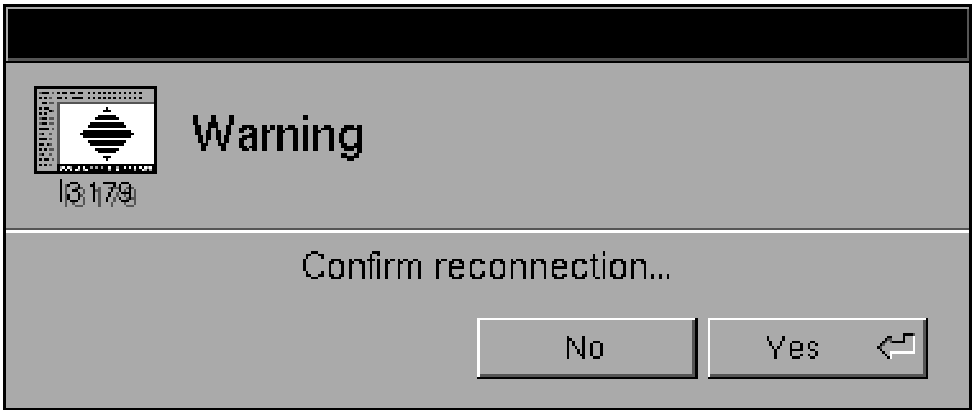




Because a “Reconnect” will first close the current session with the host (if still opened), you will get the panel below which will give you a **last** chance to cancel this action.



WE-I3179
Emulation
Confirm reconnection



Warning

Confirm reconnection...

4.4 Starting the communication processor simulator

This is only for a demonstration of the WE-I3179 terminal emulator if you have no IBM™ connection available. This may also be used to check the correctness of your installation independently of any IBM™ connection.

In the subdirectory <demo>, you will find the simulator program <we-i3179-host>. When you are done with the installation and configuration of the WE-I3179 product, you may start it for a test with the following command:

```
we-i3179-host SymFileName MainframeService Type
```

1. *SymFileName*

   File name of the datafile to be used by the <we-i3179-host> program. You can choose any of the <.sym> files found in the <demo> subdirectory.

2. *MainframeServic*e

   The name of the mainframe service it must be the same name as in the parameter Mainframe_Service of the <we-i3179-config> and </etc/services> files.

3. *Type*

   Will define the type of connection between WE-COMD and WE-I3179-host. This **must** match with <type> chosen for the WE-COMD server component. Possible values for type are:

   tn3270

   Example:

```
machine% we-i3179-host I3179Demo1.sym tn3270
```

   will start the simulator with the demo file <I3179Demo1.sym> in <tn3270> communication mode.

REMARK: Please look at the README file of the <demo> subdirectory for the latest informations about <we-i3179-host>.

5 Hints about locking config file(s)

If you intend to run many WE-I3179 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 config 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.

We recommend that you put session specific parameter(s) in that configuration file and use <Lock_Setup = 1>. If you do so, WE-I3179 will lock that file at start-up time for the whole duration of the 3270 session. If you try to start a second window with the same parameter file, you will get a message 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 config file.

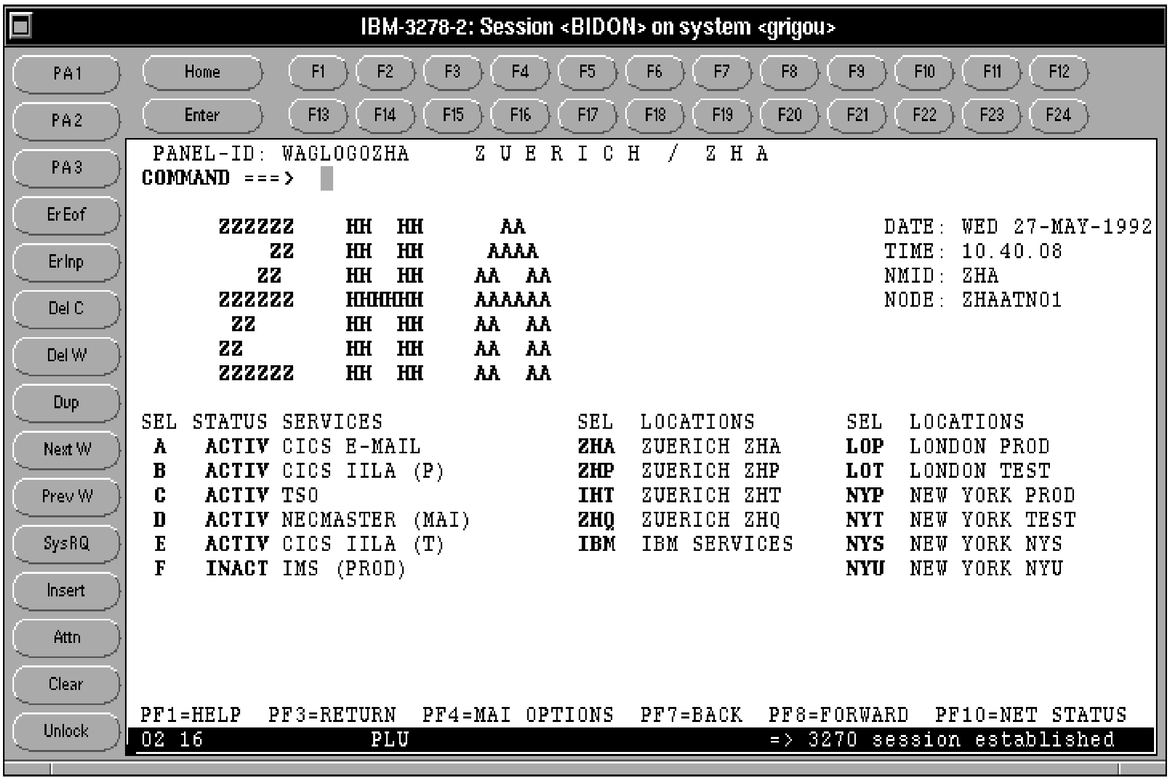
6 Using WE-I3179 terminal emulator

6.1 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-I3179 and the host connection is successful, the following window appears (data will probably be different):

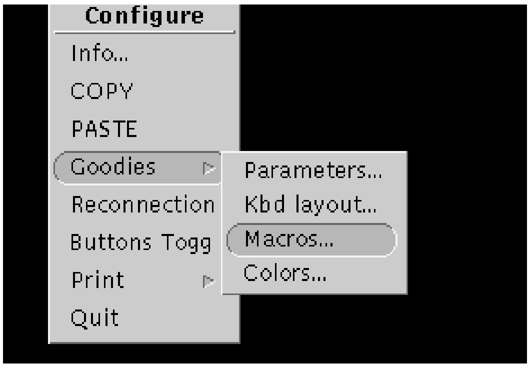


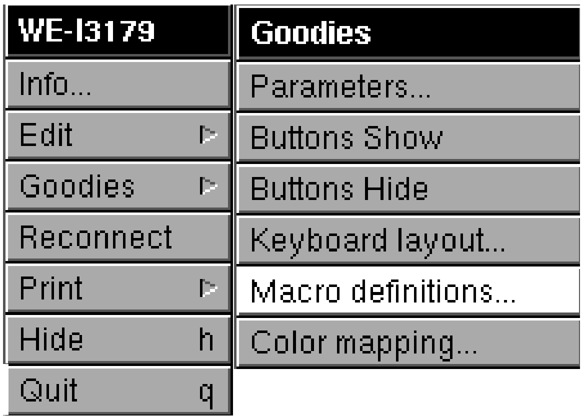
```
 PANEL-ID: WAGLOGOZHA     Z U E R I C H  /  Z H A
COMMAND ===> 

      ZZZZZZ    HH  HH      AA                              DATE: WED 27-MAY-1992
          ZZ    HH  HH     AAAA                             TIME: 10.40.08
         ZZ     HH  HH    AA  AA                            NMID: ZHA
      ZZZZZZ    HHHHHH    AAAAAA                            NODE: ZHAATN01
       ZZ       HH  HH    AA  AA
      ZZ        HH  HH    AA  AA
      ZZZZZZ    HH  HH    AA  AA

SEL  STATUS SERVICES                 SEL  LOCATIONS       SEL  LOCATIONS
 A   ACTIV CICS E-MAIL               ZHA  ZUERICH ZHA     LOP  LONDON PROD
 B   ACTIV CICS IILA (P)             ZHP  ZUERICH ZHP     LOT  LONDON TEST
 C   ACTIV TSO                       IHT  ZUERICH ZHT     NYP  NEW YORK PROD
 D   ACTIV NECMASTER (MAI)           ZHQ  ZUERICH ZHQ     NYT  NEW YORK TEST
 E   ACTIV CICS IILA (T)             IBM  IBM SERVICES    NYS  NEW YORK NYS
 F   INACT IMS (PROD)                                     NYU  NEW YORK NYU




PF1=HELP  PF3=RETURN  PF4=MAI OPTIONS  PF7=BACK  PF8=FORWARD  PF10=NET STATUS
02 16            PLU                                 => 3270 session established
```

6.2 Main menu

The two pictures on page 30 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.

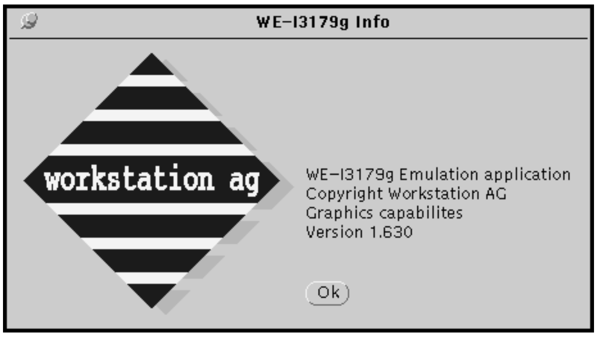
Activating “Quit” leaves the terminal emulator.





6.2.1 Info

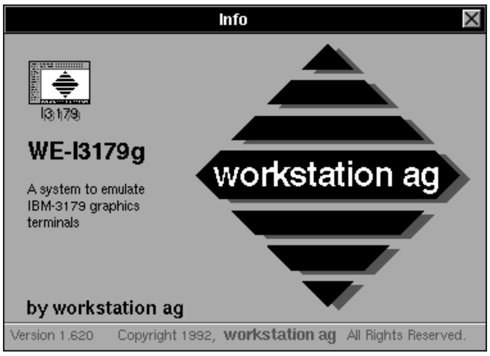
Will display an information panel containing the software release, copyright information, etc.



WE–I3179g Info

WE–I3179g Emulation application
Copyright Workstation AG
Graphics capabilites
Version 1.630

Ok



Info

I3179

WE-I3179g

A system to emulate
IBM-3179 graphics
terminals

by workstation ag

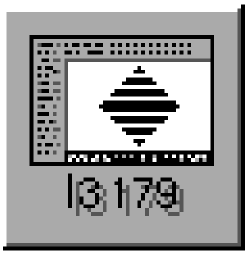
Version 1.620 Copyright 1992, workstation ag All Rights Reserved.

6.2.2 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. Under NEXTSTEP, Copy and Paste are in the Edit submenu.

6.2.3 Hide (NEXTSTEP only)

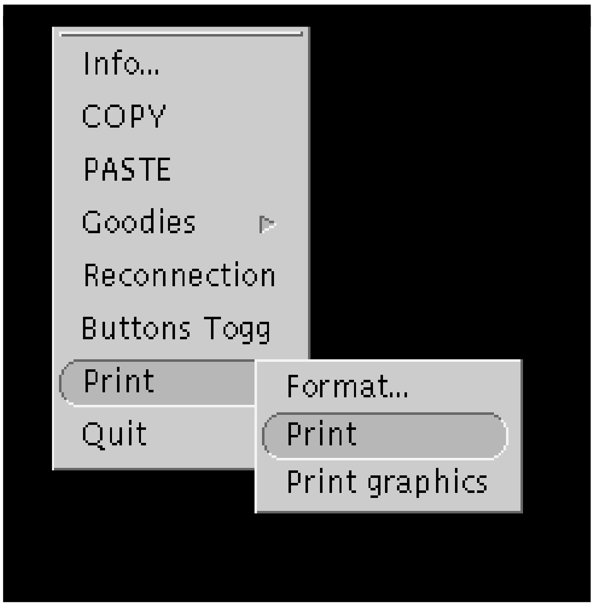
If clicked, any window or panel as well as the Main menu panel will be removed from your screen and only the I3179 icon will remain.

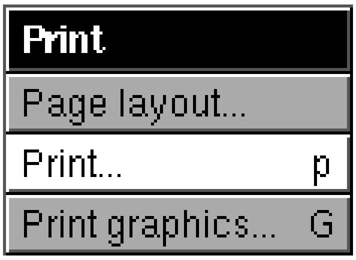


We recommend **not** to use the Hide menu function but to iconify the emulation main window by clicking in the upper left button (see Appendix D on page 79).

6.2.4 Print

When you choose this option, the following submenu appears:





Print and print graphics

Will make a hardcopy of the text or graphic (WE-I3179g only) window respectively. The format of the printout is defined by:

-> Values set by the config file(s) read at WE-I3179 start-up (see Appendix A on page 57).

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

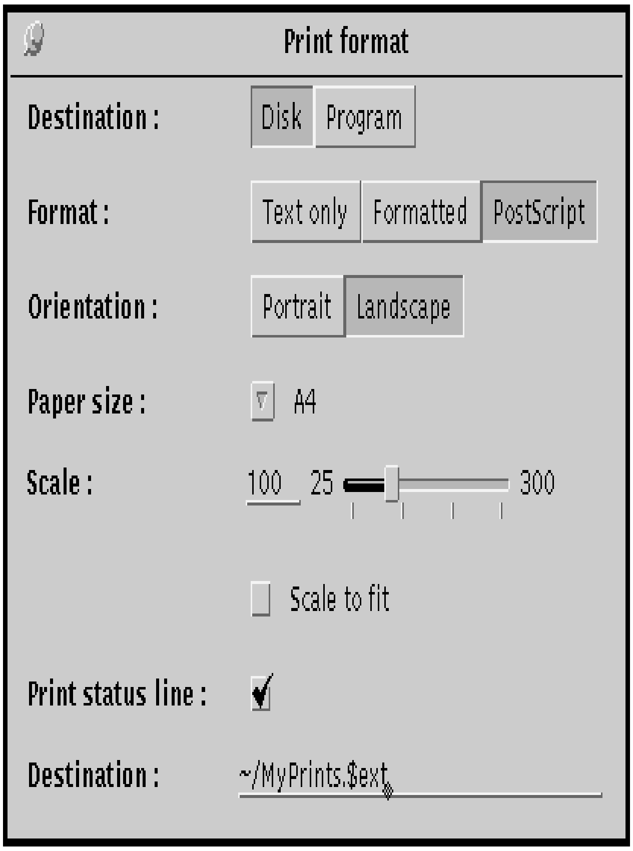
Format (X) **or page layout** (NEXTSTEP)

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:



Destination : Disk

Format : PostScript

Orientation : Landscape

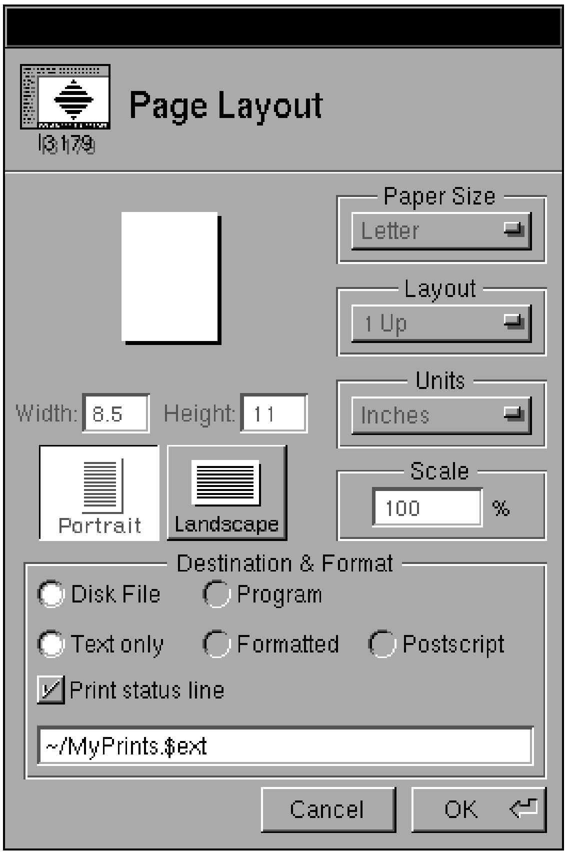
Paper size : A4

Scale : 100 (25 – 300)

□ Scale to fit

Print status line : ☑

Destination : ~/MyPrints.$ext



Paper Size: Letter

Layout: 1 Up

Units: Inches

Width: 8.5 Height: 11

Scale: 100 %

Destination & Format

○ Disk File ○ Program

○ Text only ○ Formatted ○ Postscript

☑ Print status line

~/MyPrints.$ext

At program start-up, these panels contain defaults or values set by the corresponding config (parameter) file(s) entries (see Appendix A on page 57).

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-I3179 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-I3179.

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-I3179 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 config file entries:

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

-> Program

WE-I3179 will create temporary disk files (/tmp/we-i3179/$ext) containing the print data in the given format for the meaning of $ext (see “Destination” on page 32). The text field at the panel bottom contains the name of the program to which WE-I3179 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-I3179 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 config file entries:

Print_Destination	*program*
Disk_Print	*pr -r -h $File*

Format

-> Text only

WE-I3179 will create hardcopies containing exclusively standard ASCII characters. This means that attributes and foreign characters will not be rendered.

-> Formatted

Currently the same as Text only.

-> PostScript

WE-I3179 will create hardcopies which will be an exact image of the current text or graphic window (depending if you click print or print graphics). If Print_Status_Line is **on**, the status line and a window border will be printed as well.

NOTE: Printouts of the graphic window (WE-I3179g only) may only be done in PostScript format.

REMARK: Defaults come from the following config 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-I3179 to generate the proper PostScript code according to the paper size, orientation and scale chosen.

REMARK: Defaults come from the following config 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.

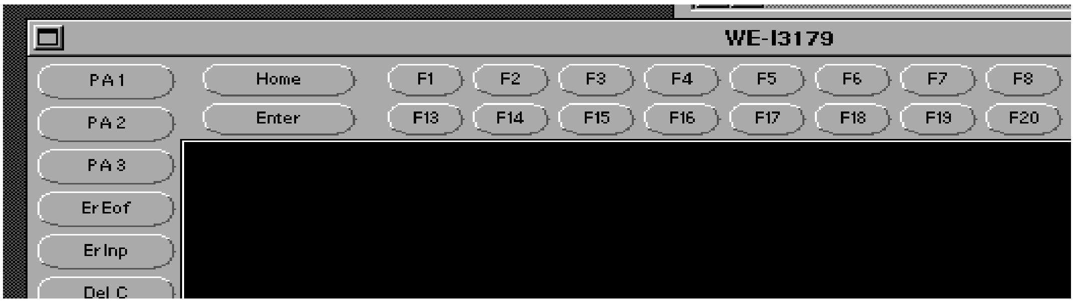
REMARK: Default comes from the following config file entries:

Print_Status_Line *0*

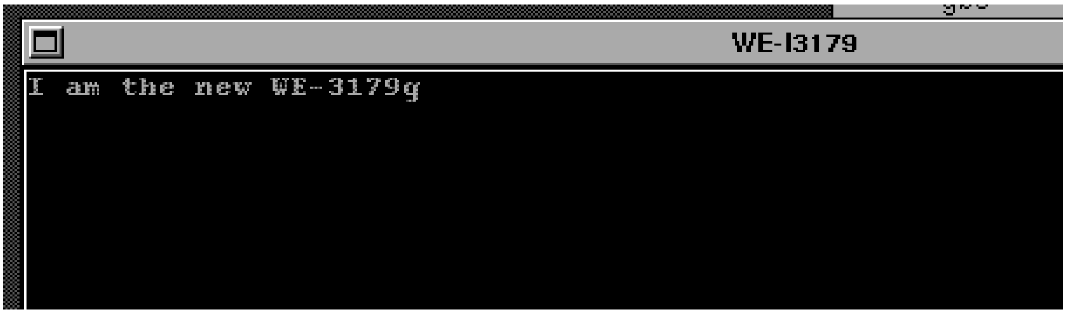
6.2.5 Buttons Show, Buttons Hide and Buttons Toggle

`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



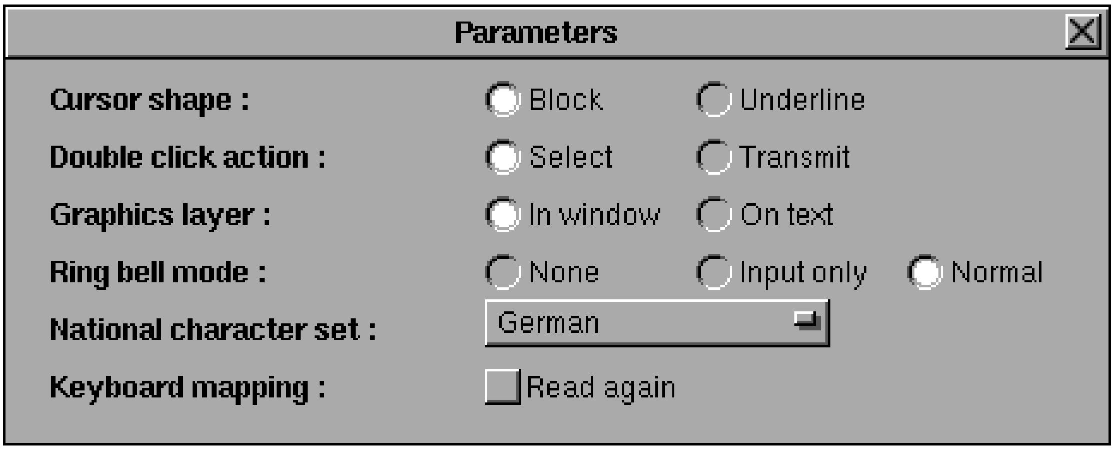
2. Example of window without buttons



```
I am the new WE-3179g
```

6.2.6 Parameters

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



Parameters ☒

Cursor shape : ● Block ○ Underline

Double click action : ● Select ○ Transmit

Graphics layer : ● In window ○ On text

Ring bell mode : ○ None ○ Input only ● Normal

National character set : German

Keyboard mapping : □ Read again

<u>**Cursor shape**</u>

This entry allows you to choose between a “Block” and an “Underline” cursor.

REMARK: At start-up time, the value is set from the config parameter:

```
Cursor_Kind          0       # Block cursor (default)
                     1       # Underline cursor
```

<u>**Double click action**</u>

This entry allows you to define what will happen when you make a double click with your mouse in the emulator window.

REMARK: At start-up time, the value is set from the config parameter:

```
Double_Click_Action  0       # Select word (default)
                     1       # Position cursor and Enter (send data to host)
```

<u>**Graphics layer**</u>

This entry allows you to choose where graphic data will be displayed (WE-I3179g only).

- In window means that graphics will be displayed in a separate window created dynamically while graphic data arrives. This option allows you to resize graphics separately and to retain the last graphic displayed while continuing to work in the text window.

- On text means that graphics will be displayed in the same window as text data, allowing a superposition of alphanumeric and graphic data. This is how an original (IBM ™) 3179g terminal displays graphic data.

- As a special feature, WE-I3179 allows you to switch back and forth between the two options (even with a graphic displayed) merely by clicking on the corresponding button.

REMARK: At start-up time, the value is set from the config parameter:

```
Graphics_layer      0     # In window (default)
                    1     # On text
```

Ring bell mode

This entry allows you to choose when WE-I3179 should beep. You have the choice between three options:

-> None: WE-I3179 will never bell.

-> Input only: WE-I3179 will bell only if you make input errors (i.e.: you try to enter data in a protected field or you press Enter or a PF/PA key while WAIT is displayed in the status line).

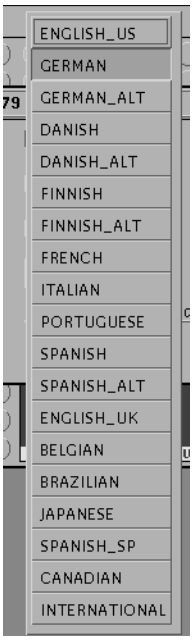
-> Normal: This mode is the same as Input only. In addition, bell actions requested by the IBM™ host (sound alarm bit in WCC) are also honored.

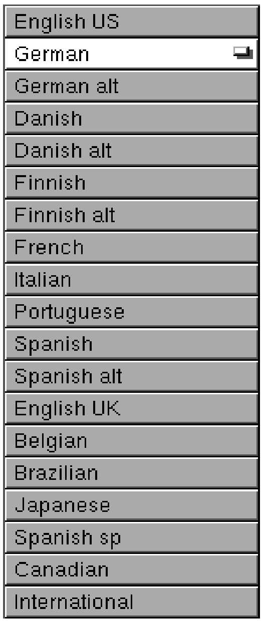
REMARK: At start-up time, the value is set from the config parameter:

```
Bell_Value      0     # None
                1     # Input only
                2     # Normal (default)
```

National character set

With the pull down menu you can choose the national character set.





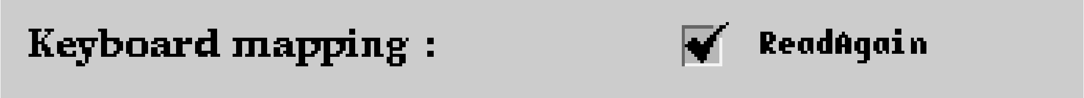
-> International:	The international character set (CECP International 5) will be used. This also corresponds to the new code pages 500/1 which is a superset of code page 908.
-> All others:	The corresponding national character set will be used.

REMARK: At start-up time, the value is set from the config parameter:

```
National_Cset    ENGLISH_US
                 GERMAN
                 ...
```

Keyboard mapping

This entry allows you to reread the keyboard mapping file without leaving the WE-I3179 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 7 on page 47.



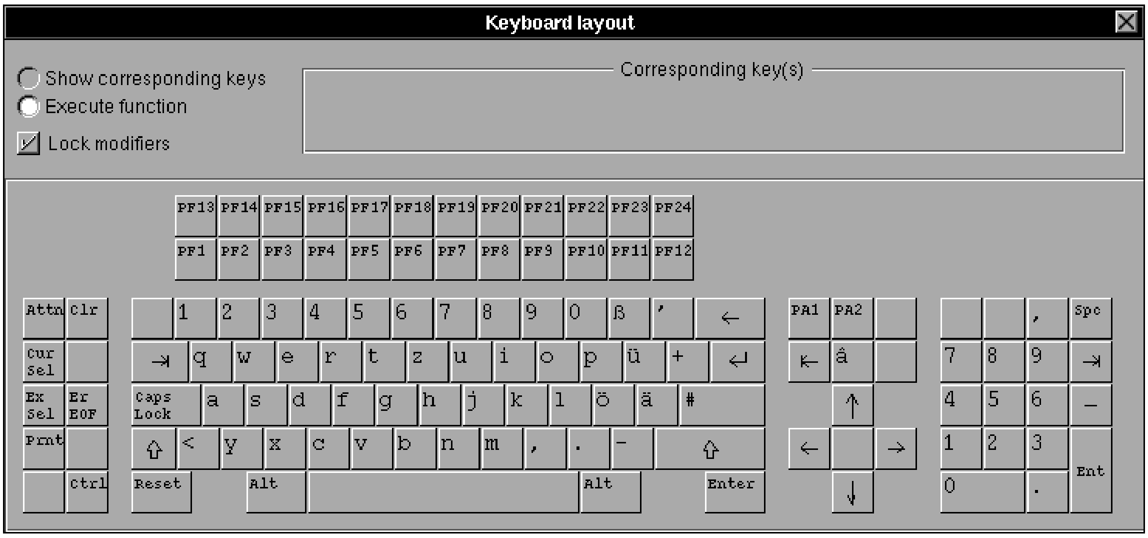
Keyboard mapping : ☑ ReadAgain

REMARK: At start-up time, the file from which the keyboard mapping comes is defined with the “Keymap_File” option as follows:

```
Keymap_File    /LocalApps/we-i3179.app/US.I3179_keymap
```

6.2.7 Keyboard layout

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



Keyboard layout

○ Show corresponding keys

○ Execute function

☑ Lock modifiers

Corresponding key(s)

PF13 PF14 PF15 PF16 PF17 PF18 PF19 PF20 PF21 PF22 PF23 PF24
PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12
Attn Clr
Cur Sel
Ex Sel Er EOF
Prnt
Ctrl
1 2 3 4 5 6 7 8 9 0 ß ’ ←
⇥ q w e r t z u i o p ü + ↵
Caps Lock a s d f g h j k l ö ä #
⇧ < y x c v b n m , . - ⇧
Reset Alt Alt Enter
PA1 PA2
⇤ â
↑
← →
↓
, Spc
7 8 9 ⇥
4 5 6 _
1 2 3 Ent
0 .

The picture on page 38 represents an original 3270 keyboard. 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.: Enter, Tab, PF6, etc.). Beside your workstation's physical keyboard and the buttons around the emulator window, this is a third way to enter data into your IBM™ emulation window.

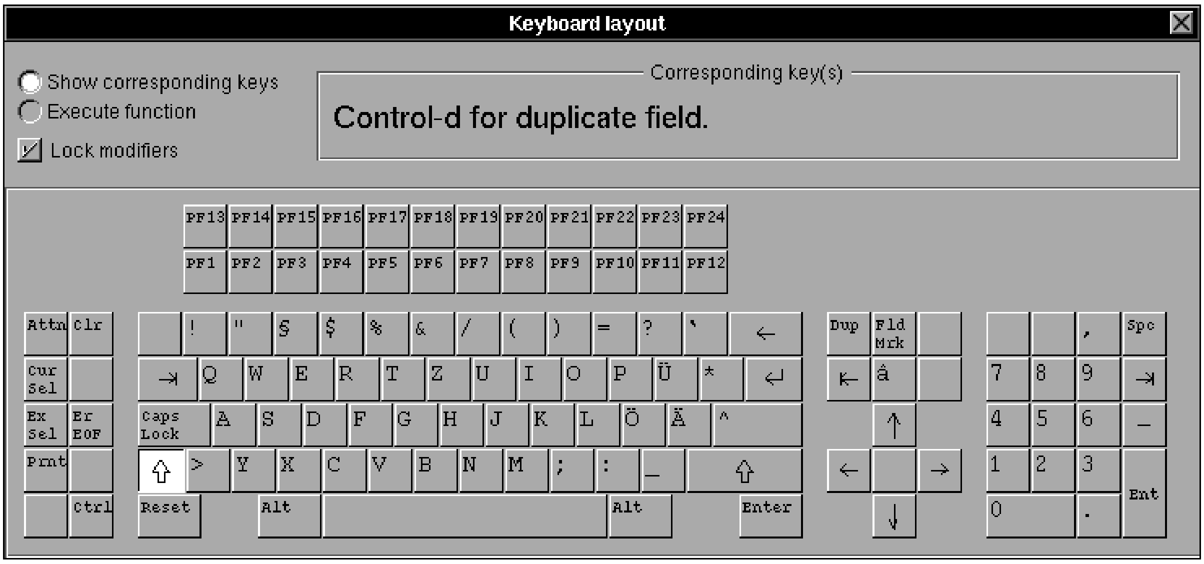
Show corresponding keys mode

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 3270 keyboard displayed here.

The assignment of a key, a combination or a sequence of keys to a particular 3270 function is done with the ***keymapper tool*** described in chapter 7 on page 47. 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 on page 40). 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.



Keyboard layout

○ Show corresponding keys
○ Execute function
☑ Lock modifiers

Corresponding key(s)

Control-d for duplicate field.

This picture shows you a typical show mode output. The user clicked on the <Dup> key and the following line was displayed:

Control-d for duplicate field

=> This is the comment field which was entered with the ***keymapper tool***.

NOTE 1: In fact, when the user presses “Ctrl d” together, some code (i.e.: “ctrl d”) will be sent from your window server to the WE-I3179 emulation program which, in turn, will execute the dup function.

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 config file for the name of the keyboard mapping file (the <Keymap_File> entry). For our example, we will assume that it’s name is <km.dat>.

2. Start an emulation with this config file and connect to the IBM™ mainframe.

3. Start a keymapper tool (KM-I3179) 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.

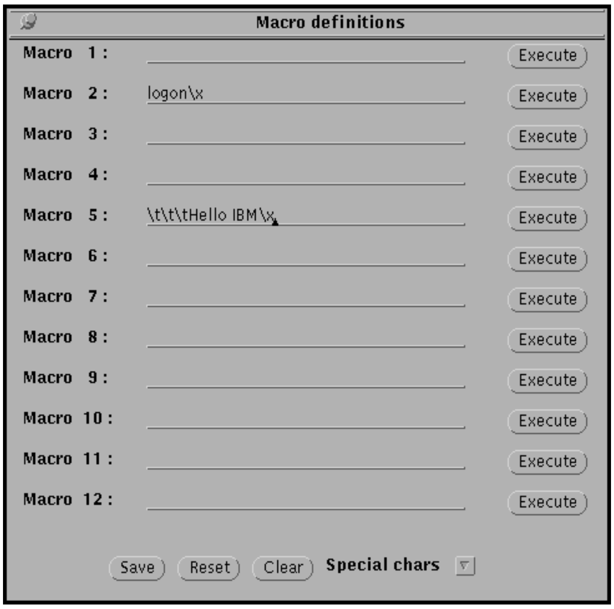
– In step 4, you should comment all entries with a sentence stating clearly what keys are to be depressed to produce the corresponding 3270 function.

– If you don’t know anymore which key combination you have assigned to a 3270 function, display the “keyboard layout” (Goodies), and use it in show mode.

NOTE: For more informations about keyboard mapping, please refer to chapter 7 on page 47.

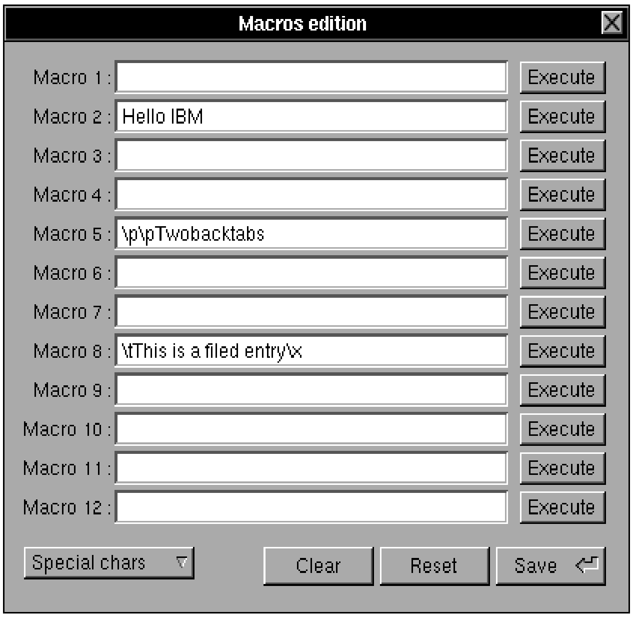
6.2.8 Macro definitions

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



Macro 2 : logon\x

Macro 5 : \t\t\tHello IBM\x



Macro 2 : Hello IBM

Macro 5 : \p\pTwobacktabs

Macro 8 : \tThis is a filed entry\x

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

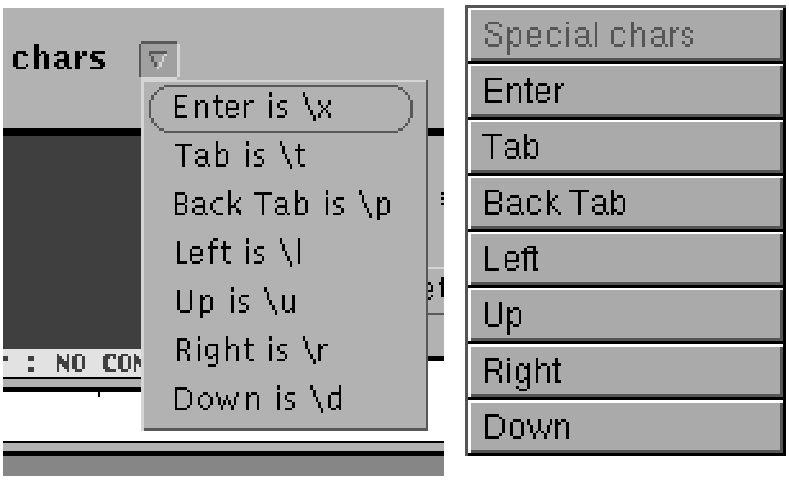
When clicked, this will revert **all** macro entries to the last saved version.

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 5 (when activated) make 3 Tabs, put Hello IBM and XMIT these data to the host.

1. Click at the beginning of the macro 5 field with the mouse.

2. For NEXTSTEP:
   With the mouse, click on special chars and select the “Tab” entry. When selected, release the mouse button. This will automatically enter a “\t” entry at the beginning of the macro 5 field. Make it 3 times for the 3 Tabs.

   For X windows:
   With the mouse, click on special chars and read the indication on the right of the “Tab is” entry. Then, enter the value read (\t) 3 times into the macro 5 field.

3. Enter Hello IBM in the macro 5 entry after the “\t\t\t”.

4. Repeat step 2 but choose the <Enter> entry.

5. Try this macro by clicking on the Execute button sitting on the right of the macro 5 entry.

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 47 and enter key combinations for the M_MACRO1EXEC to M_MACRO12EXEC entries.

6.2.9 Color mapping

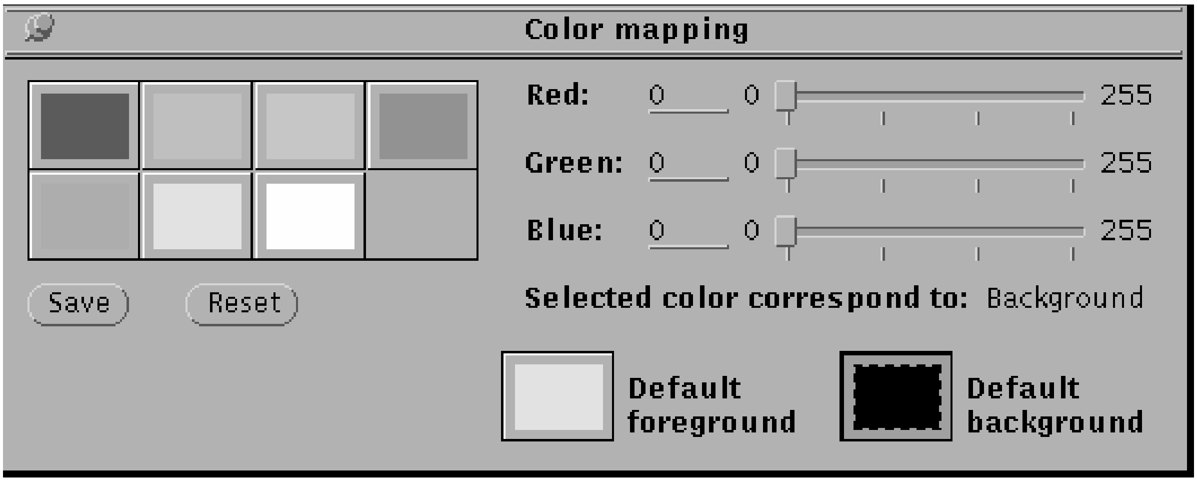
Purpose of the color mapping utility is to allow the user to change the original 3270 colors to your preferred colors. The 3270 protocol specifies 7 colors which can be mapped. You may also change the mapping of the background and the foreground colors.

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

REMARK: The foreground color applies only to the last window line (status line).

6.2.10 Color mapping under X

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



Color mapping

Red: 0 0 255

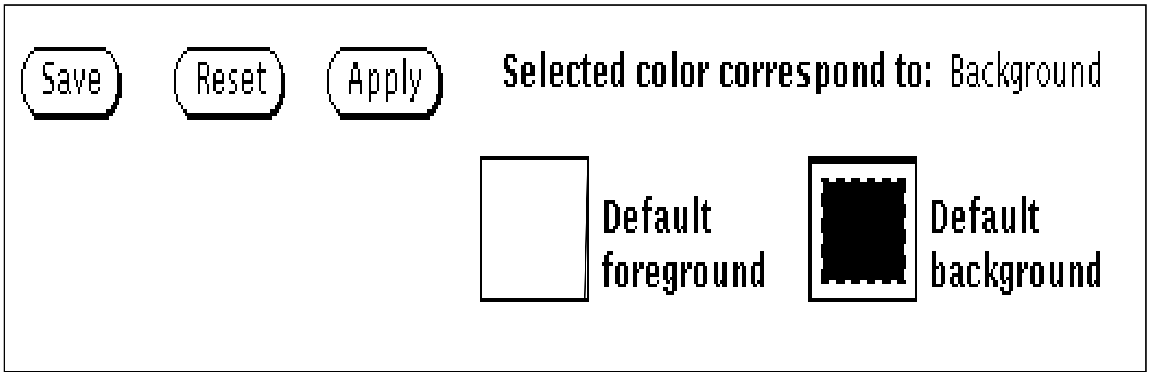
Green: 0 0 255

Blue: 0 0 255

Selected color correspond to: Background

Color mapping panel for X window systems (Dynamic Color Map)

If your X server has a Static Color Map, the lower part of the color mapping panel (see picture on page 43) will look as below:



WE-I3179 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 mainframe and display a colorful screen.
2. In the Goodies menu, select color mapping. One of the color editing tool panel shown above 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:

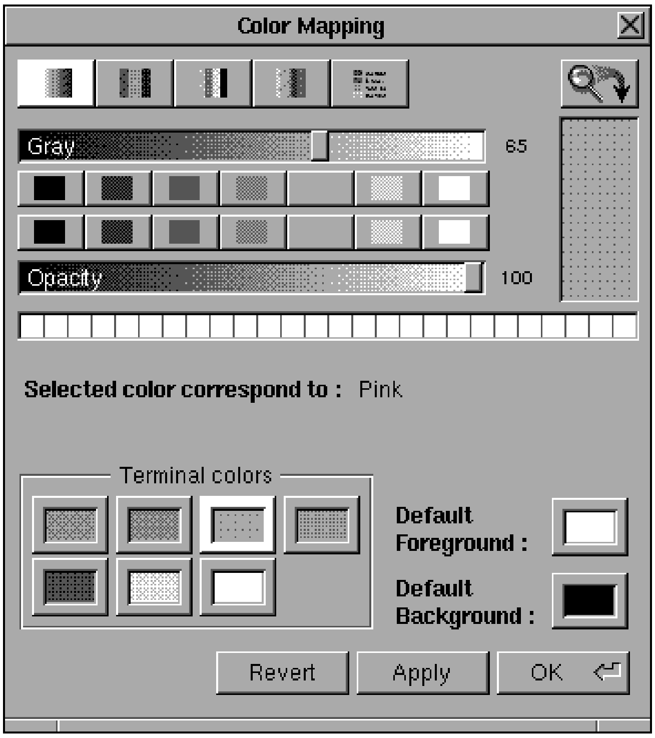
   -> For *Dynamic Color Maps*, the result will be immediately visible in the color mapping tool and in your emulator character window while moving the slider.

   -> 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).

6.2.11 Color mapping under NEXTSTEP

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



Selected color correspond to : Pink

NEXTSTEP color mapping tool for WE-I3179 (NEXTSTEP Release 2.0)

REMARK: WE-I3179 uses a NeXT object for the color edition. This object may be improved from one NEXTSTEP release to another and WE-I3179 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.

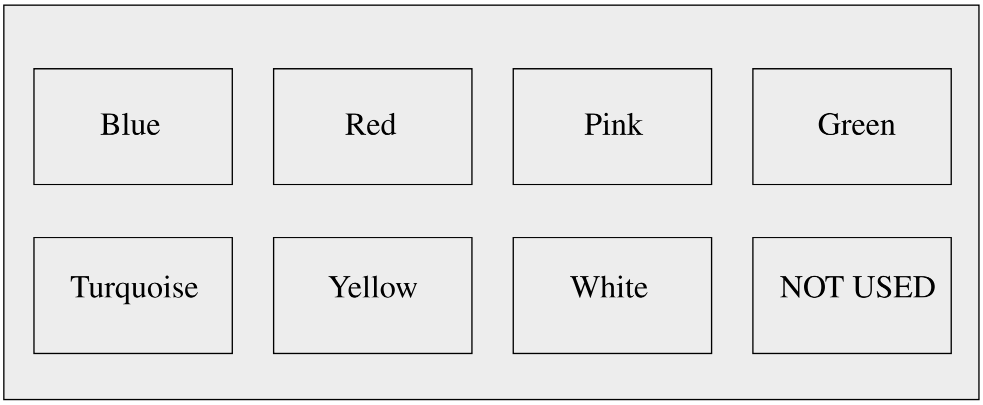
Revert button

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).

Correspondence between 3270 colors and mapping tool colors



How to proceed

1. Connect to the mainframe and display a colorful screen.

2. In the Goodies menu, select color mapping. One of the color editing tool panel shown above 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 one of the seven 3270 colors at the bottom of the panel or 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.

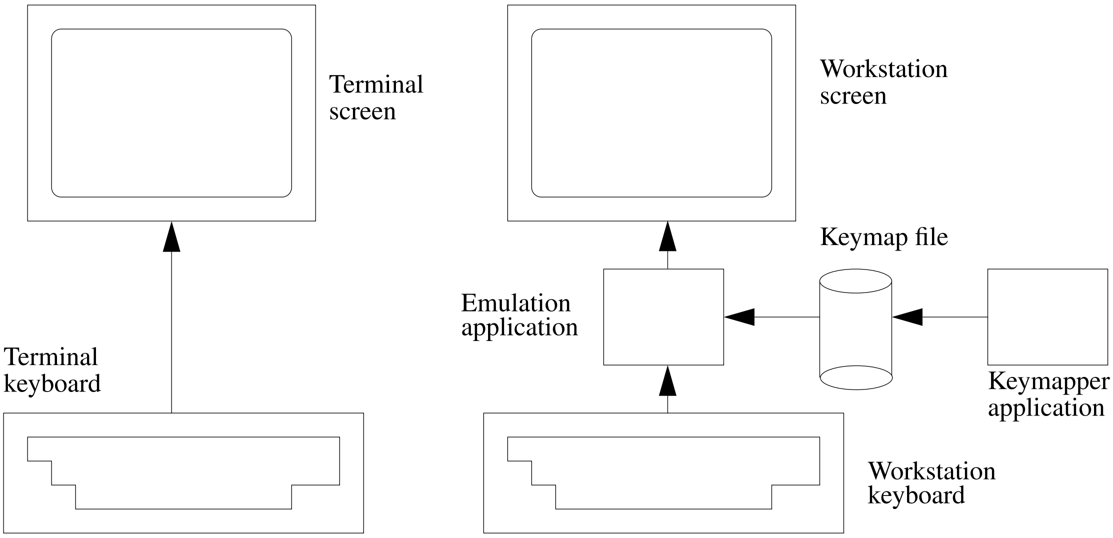
4. When you are done, click the OK button to save your work in the file indicated with <Color_File> in the active configuration (i.e.: color.dat). The color mapper tool panel will be closed.

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.

7 Keymapper tool

7.1 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.

7.2 Definitions

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

7.2.1 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”.

7.2.2 Terminal keys

are alphabetical, numeric and other ASCII symbol keys.

7.2.3 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.

7.2.4 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.

7.2.5 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.

7.2.6 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, ˆ, Ç, etc.). Special keys have names, always starting with a slash (‘/’); they are listed in chapter 7.7 on page 54.

7.2.7 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.

7.3 Installing and starting the keymapper tool

7.3.1 Foreword

This chapter is common for the whole WE-XXX product family. At present time, 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.

7.3.2 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 in 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 53).

To be recognized as the document of the keymapper application (visualized when 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.

7.3.3 Under X windows

Copy all the files from the distribution media into a directory, and type the following 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:

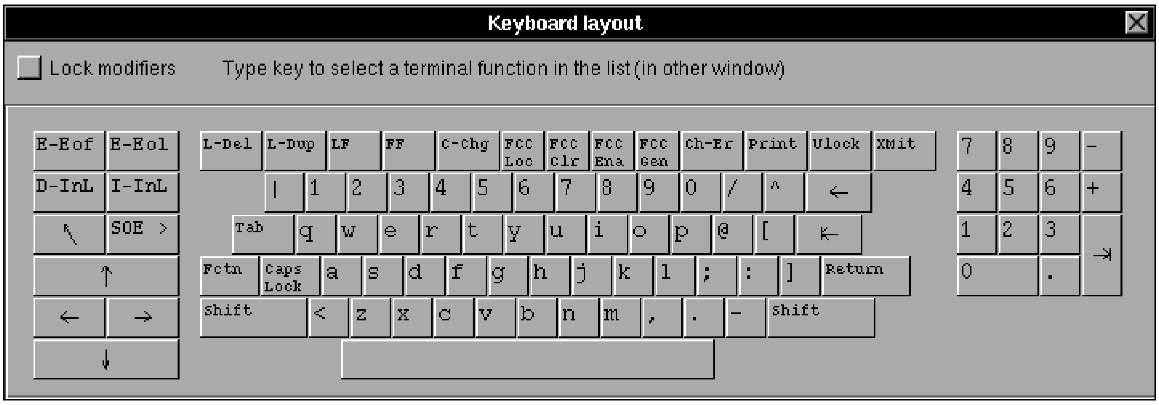
km-uts	for the WE-UTS emulation
km-d320	for the WE-D320 emulation
km-i3179	for the WE-I3179 emulation

Then, proceed with the next chapter.

7.4 Using the keymapper tool

7.4.1 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.



Keyboard layout

□ Lock modifiers

Type key to select a terminal function in the list (in other 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.

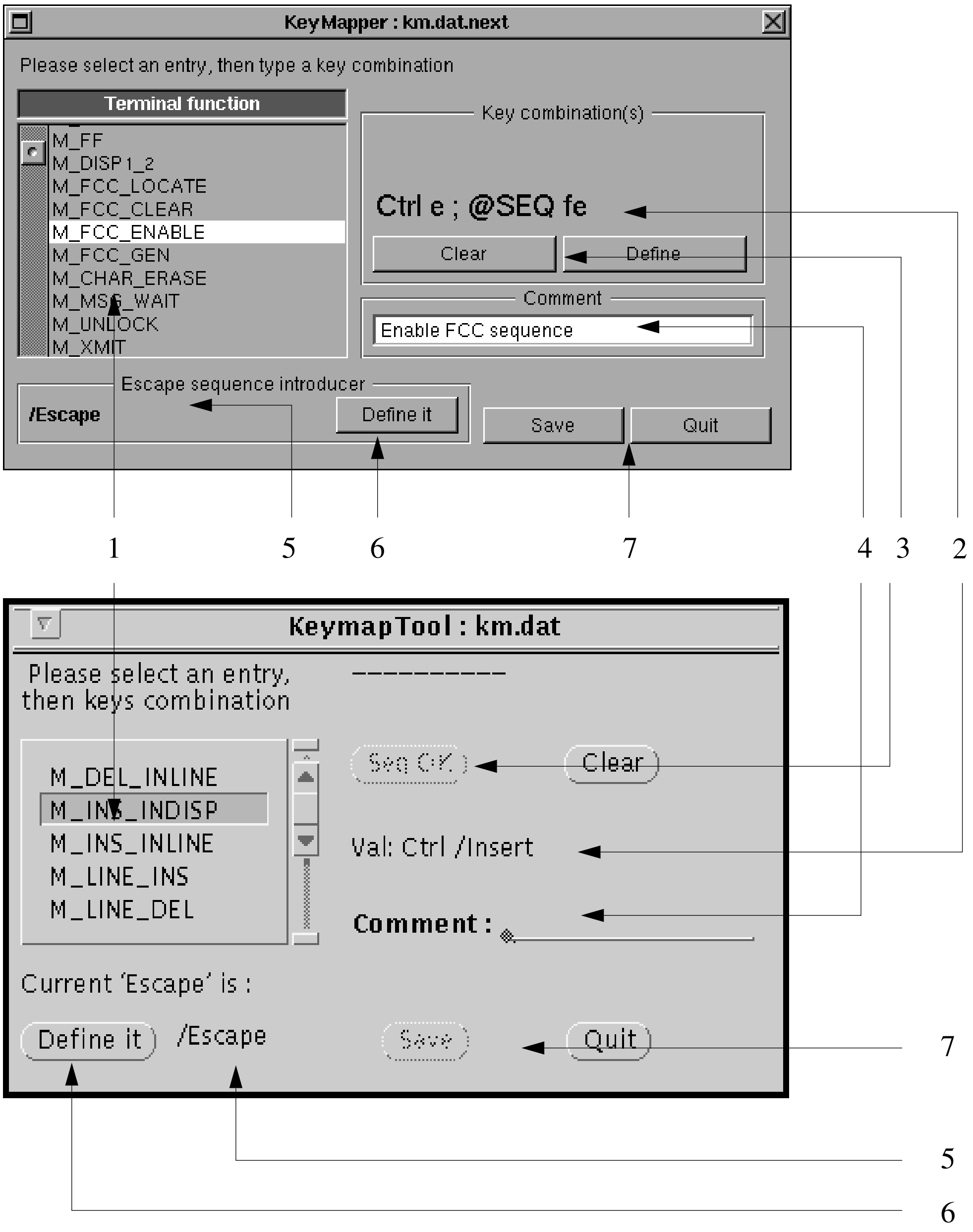
NOTE 2: (For NeXT users only): You cannot use the “Command” modifier in the key combinations you define, because “Command” should be used only for menu shortcuts (accelerators), and terminal functions are not provided in the application menus.

The illustrations on page 51 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.



KeyMapper : km.dat.next

Please select an entry, then type a key combination

Key combination(s): Ctrl e ; @SEQ fe

Comment: Enable FCC sequence

Escape sequence introducer: /Escape

KeymapTool : km.dat

Please select an entry, then keys combination

Val: Ctrl /Insert

Comment :

Current ‘Escape’ is : /Escape

(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.

7.4.2 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.

7.4.3 Key combination field

A terminal function or key can be associated with 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 7.7 on page 54). 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.

7.4.4 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 behavior 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).

7.4.5 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.

7.4.6 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.

7.4.7 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 1 sequence intro key.

7.4.8 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.

7.5 Tips for advanced users

7.5.1 Command line parameters

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

Allow 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.

7.6 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.

7.7 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	÷

Appendix A

A WE-I3179 parameter file format

```
# WE-I3179 WINDOW PARAMETERS
Appl_Name            3090                  # Session name in upper window border and icon
Point_Mouse          1                     # We will be able to position cursor with mouse
Buttons_Panel        1                     # Window will be bordered with buttons
Exit_Confirm         1                     # A panel will be displayed before exit
Lock_Setup           0                     # No locking of primary configuration file
Bell_Value           2                     # Bell if input errors or WCC sound alarm bit set
Double_Click_Action  1                     # Set cursor and enter (will send data to host)
Cursor_Kind          0                     # Block cursor

# WE-I3179 BLINKING OPTIONS
Blink_Cursor         1                     # Cursor will blink
Blink_OutFocus       1                     # And continue to blink while window not selected
Inactiv_Time         15                    # Stop any blink after 15 second inactivity

# WE-I3179 TERMINAL TYPE OPTIONS
Term_Name            IBM-3278              # Will be snet through TELNET to IBM host
Term_Type            2-E                   # 24x80 terminal with extended highlighting
National_Cset        INTERNATIONAL         # Code page 500 will be used

# WE-I3179 CHARACTER FONTS (used by WE-I3179c and WE-I3179g)
Font_R               Courier               # Good for NEXTSTEP
Font_B               Courier-Bold          # Good for NEXTSTEP
Font_Dimensions      8,10,12,14,18,19,24   # Needed by X servers only
Start_Height         12                    # Will give window start-up size
Font_P               Helvetica             # Good for NEXTSTEP
Font_Warning         1                     # Yes we want to know about font errors
Invert_Bold          0                     # Don’t invert normal and bold attributes

# WE-I3179 GRAPHIC LAYER (used by WE-I3179g only)
Graphic_Layer        0                     # Put graphics into separate window

# WE-I3179 GRAPHIC FONT AND SIZE (used by WE-I3179g only)
Graphics_Font        courb.pfa             # Will use courier (bold) for graphics
Graphics_Width       720                   # Width of the graphic window
Graphics_Height      384                   # Height of the graphic window

# WE-I3179 SERVER AND HOST CONNECTION OPTIONS
Comd_Host            localhost             # Will try to connect to a local we-comd
Comd_Service         wag3270               # Will be used between we-i3179 and we-comd
Mainframe_Host       MyIbm3090             # Internet name of IBM host (or controller)
Mainframe_Service    telnet                # Service used between we-comd and IBM
```

```
# WE-I3179 AUXILIARY FILES
Keymap_File          km.dat                      # Where keymapping is stored
Macro_File           macro.dat                   # Where macro lives
Kbd_Layout_File      keylay.dat                  # Original IBM keyboard image file
Color_File           color.dat                   # Where our custom colors are stored
Buttons_File         button.dat                  # Your custom buttons definitions

# WE-I3179 PRINT PARAMETERS
Disk_Print           ~/we-i3179.$ext             # Destination for disk print
Print_Program        lpr -r -h $File             # Destination for program print
Print_Dest           Disk                        # Destination is DISK
Print_Status_Line    1                           # Add status line and border
Print_Format         Postscript                  # Format is PostScript
Print_Output_Size    A4                          # (PostScript only) Paper size
Print_Orientation    Landscape                   # (PostScript only) Orientation
Print_Scale          80                          # (PostScript only) Scaling
Print_Generic_File   ~/we-i3179/ps_generic.dat   # (PostScript on X only) Skeleton

# WE-I3179 PASSWORD
Pass_Word            PutYourPasswordHere         # Your licence
```

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 16).

A.1 Parameter description

REMARK: The entries marked with an * can be modified at run time by clicking in the WE-I3179 parameter panel.

In all entries containing a file name, you may place a ~ at the beginning of the path. WE-I3179 will replace it automatically by the user's home directory (i.e.: `Color_File` ~/we-i3179/color.dat).

1. Appl_Name

– Definition: This entry has NO functionality. It is merely used to display an application name in the upper window border (while connected to the IBM host) and is also displayed in the icon while the WE-I3179 is iconified.

– Defaults: None

– Mandatory: No

2. Point_Mouse

– Definition: If set to 1 (YES), the left mouse button in the WE-I3179 window will place the cursor on the position pointed to or to the next unprotected screen position.

– Defaults: 0 (NO)

– Mandatory: No

3. Buttons_Panel *

– Definition: If set to 1 (YES), clickable buttons representing the special keys of an WE-I3179 terminal (i.e.: functions keys) will be displayed around the WE-I3179 terminal window. You will then be able to activate all these special functions by clicking them with the mouse.

– Defaults: 0 (NO)

– Mandatory: No

NOTE: At any time during the WE-I3179 session you can toggle this option using the WE-I3179 window menu. This option defines only the start-up value for this parameter.

4. 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 you session.

– Defaults: 0 (No)

– Mandatory: No

5. Lock_Setup

– Definition: If set to 1 (YES), the primary setup file (see “Hints about locking config file(s)” on page 27) will be locked (reserved) for the whole duration of the WE-I3179 session. You may use this option only if you have write access to the primary setup file.

– Defaults: 0 (NO)

– Mandatory: No

6. Bell_Value *

– Definition: This antry allows you to choose when WE-I3179 should beep. Possible values are:

0 (None):
WE-I3179 will never bell.

1 (Input only):
WE-I3179 will bell only if you make input errors (i.e.: you try to enter data in a protected field).

2 (Normal):
This mode is the same as mode <1>. In addition, bell actions requested by the IBM™ host (sound alarm bit in WCC) are also honored.

– Defaults: 2

– Mandatory: No

7. Double_Click_Action *

– Definition: This antry allows you to define what will happen when you make a double click with your mouse in the emulator window:. Possible values are:

0 => Select word (default)

1 => Position cursor and enter (send enter AID and data to host)

– Defaults: 0

– Mandatory: No

8. Cursor_Kind *

– Definition: This entry allows you to choose between a block and an underline cursor. Possible values are:

0 => Block cursor

1 => Underline cursor

– Defaults: 0 (Block)

– Mandatory: No

9. 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-I3179 window is out of focus (see <Blink_OutFocus> below).

– Defaults: 0 (NO)

– Mandatory: No

10. 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.

– Defaults: 1 (YES)

– Mandatory: No

11. 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

12. Term_Name

– Definition: The type given here will be used while establishing the connection with the IBM™ host (communication controller). Possible values are:

IBM-3278

– Defaults: None

– Mandatory: Yes

13. Term_Type

– Definition: The type given here will be used while establishing the connection with the IBM™ host (communication controller). Possible values are:

2	**Will be model 2 terminal (24*80**
3	**Will be model 3 terminal (32*80)**
4	**Will be model 4 terminal (43*80)**
5	**Will be model 4 terminal (27*132)**

If you want to use colors, add <-E> after the terminal model number (i.e.: 2-E)

– Defaults: None

– Mandatory: Yes

14. National_Cset *

– Definition: This is the name of the default character set that will be used for your WE-I3179 session. Possible values are:

ENGLISH_US
GERMAN_ALT
DANISH_ALT
FINNISH_ALT
ITALIAN
SPANISH
ENGLISH_UK
BRAZILIAN
SPANISH_SP
INTERNATIONAL
GERMAN
DANISH
FINNISH
FRENCH
PORTUGUESE
SPANISH_ALT
BELGIAN
JAPANESE
CANADIAN

– Defaults: INTERNATIONAL

– Mandatory: No

15. Font_R and Font_B

– Definition: These entries defines the fonts to be used for the text (data) of your WE-I3179 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 information see “Fonts specifications” on page 72.

– Defaults: None

– Mandatory: Yes

16. Font_Dimensions

– Definition: You must put here all sizes of `Font_R` and `Font_B` that you want to use in the WE-I3179 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

17. `Start_Height`

– Definition:	Here you have to put the size of `Font_R` and `Font_B` that you want to use at WE-I3179 start-up time. It must be one of the sizes listed above.
– Defaults:	None
– Mandatory:	Yes

18. `Font_P`

– Definition:	This entry defines the font to be used for the buttons in the panels (see “Buttons_Panel *” on page 59). The font given here **must** exist and **must** be accessible to your window server. For more information see “Fonts specifications” on page 72.
– Defaults:	None
– Mandatory:	Yes

NOTE: The font size chosen here will define the minimum size of the WE-I3179 window while the buttons are displayed.

19. `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.
– Defaults:	0 (No)
– Mandatory:	No

20. `Invert_Bold`

– Definition:	If you set this option to 1 (YES), the bold and normal rendition attributes will be exchanged. This means that WE_I3179 will use the `Font_R` to draw intensified fields and `Font_B` to draw the other fields.
– Defaults:	0 (NO)
– Mandatory:	No

21. Graphic_Layer

– Definition: This parameter allows you to choose if the graphics to come (WE-I3179g only) will be displayed in the text window (mix text and graphic) or in a separate window.

0 => Graphics will go into a separate window
1 => Graphics will go in the text window

– Defaults: 0 (Separate window for graphics)

– Mandatory: No

22. Graphics_Font
– Used by WE-I3179g graphic emulation only –

– Definition: The name of the scalable font used for a WE-I3179g graphic window.

Under X, you can use any of the three public domain fonts provided by workstation ag:

-> cour.pfa (standard courier font)
-> courb.pfa (courier font bold)
-> couri.pfa (courier font italic)

Under NEXTSTEP, you can use any monospaced font like:

-> Courier (standard courier)
-> Ohlfs (Ohlofs)

– Defaults: None

– Mandatory: Yes, if you want to display graphics and have purchased the according licence.

23. Graphics_Width

– Definition: The horizontal start-up size (in pixels) of an WE-I3179g graphic window.

– Defaults: 720 (same as an original 3179g)

– Mandatory: No

24. Graphics_Height

– Definition: The vertical startup size (in pixels) of an WE-I3179g graphic window.

– Defaults: 384 (same as an original 3179g)

– Mandatory: No

25. Comd_Host

– Definition: This is the name of the (UNIX) host on which the protocol server of the emulation is running. Since the “WE-COMD” component of the emulation may be loaded and running separately on each workstation of the network running WE-I3179 or be common for all, it may be necessary to specify on which (UNIX) host it is running.

– Defaults: The host on which WE-I3179 is running.

– Mandatory: No (if WE-COMD is running locally).

NOTE 1: If the protocol server is not running on the same (UNIX) host as WE-I3179, then this entry is mandatory. Otherwise, the protocol server will not be accessible and you will get an error message at WE-I3179 start-up time.

NOTE 2: If the machine on which WE-I3179 is running, is part of a YP (NIS) network, remember that “localhost” is no longer the host on which you are working, but the YP server.

26. Comd_Service

– Definition: This is the name of the service under which the “WE-COMD” component is waiting for the WE-I3179 connection requests. The name you give here **must** match the one given while starting the “WE-COMD” component and must have been declared in “/etc/services”, YP (Yellow Pages = NIS) or Netinfo (NeXT).

– Defaults: None

– Mandatory: Yes

27. Mainframe_Host

– Definition: This is the name of the (IBM™) host to which WE-I3179 will try to connect. It will also appear on the window border while the connection is established and active.

– Defaults: None

– Mandatory: No

28. Mainframe_Service

– Definition: This is the name of the service under which the "WE-COMD" component of the WE-I3179 emulation will try to connect the (IBM™) host. This must have been declared in "/etc/services", YP (Yellow Pages = NIS) or Netinfo (NeXT). Possible entries are:

telnet

– Defaults: None

– Mandatory: Yes

29. Keymap_File

– Definition: The name of the file containing the definitions for your keyboard mapping. You may modify this file using the "KM-I3179" utility. Please refer to the "Keymapper tool" on page 47 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 "Sample.I3179_keymap" with the emulation. This is a template that you must modify to match your requirements.

30. 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 41).

– 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.

31. Kbd_Layout_File

– Definition: The name of the file containing the image of the original IBM 3270 keyboard used by the "KM-I3179" utility. Can also be found in the "Goodies menu" under "keyboard layout". You can't modify this file.

– Defaults: None

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

NOTE: This entry should point to the “keylay.dat” file delivered with the application. Only the pathname may be modified.

32. 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.

33. Buttons_File

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

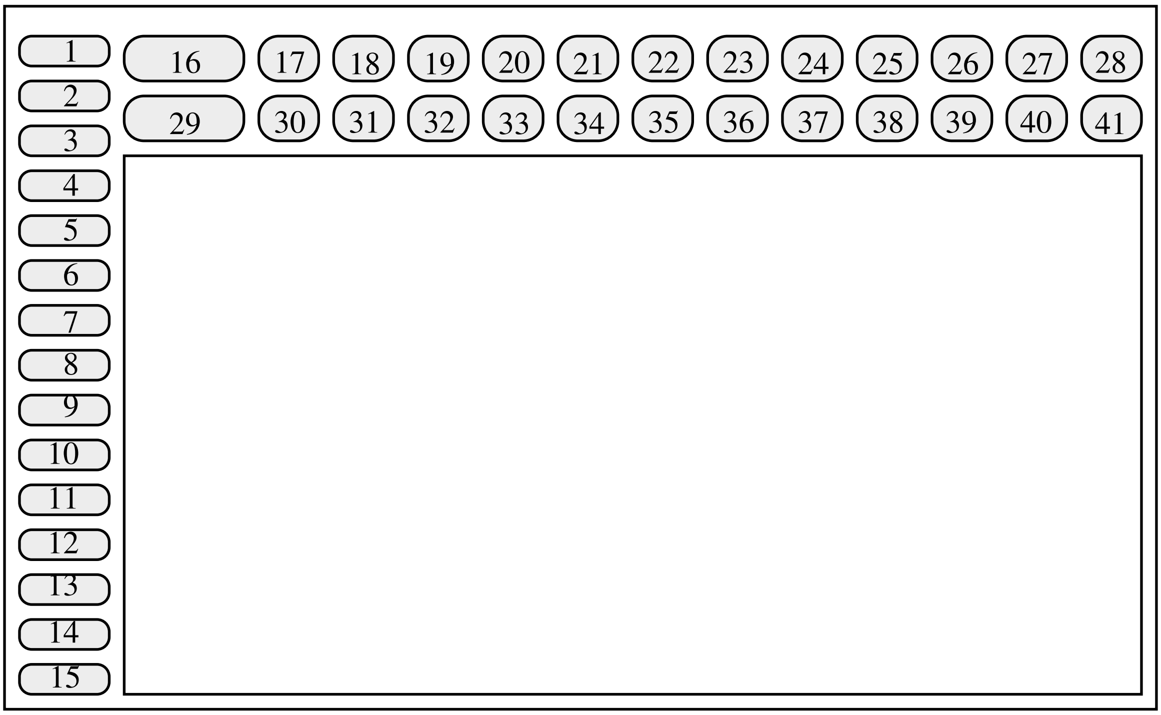
<Index, Button_title, M_func_name>
<Index, Button_title, M_func_name>
<Index, Button_title, M_func_name>
...

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 E on page 83).

– 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, “Field Mark”, M_FMARK>

Will display Field Mark in button 3. Clicking this button, will send a <Field Mark> function code to the attached application.

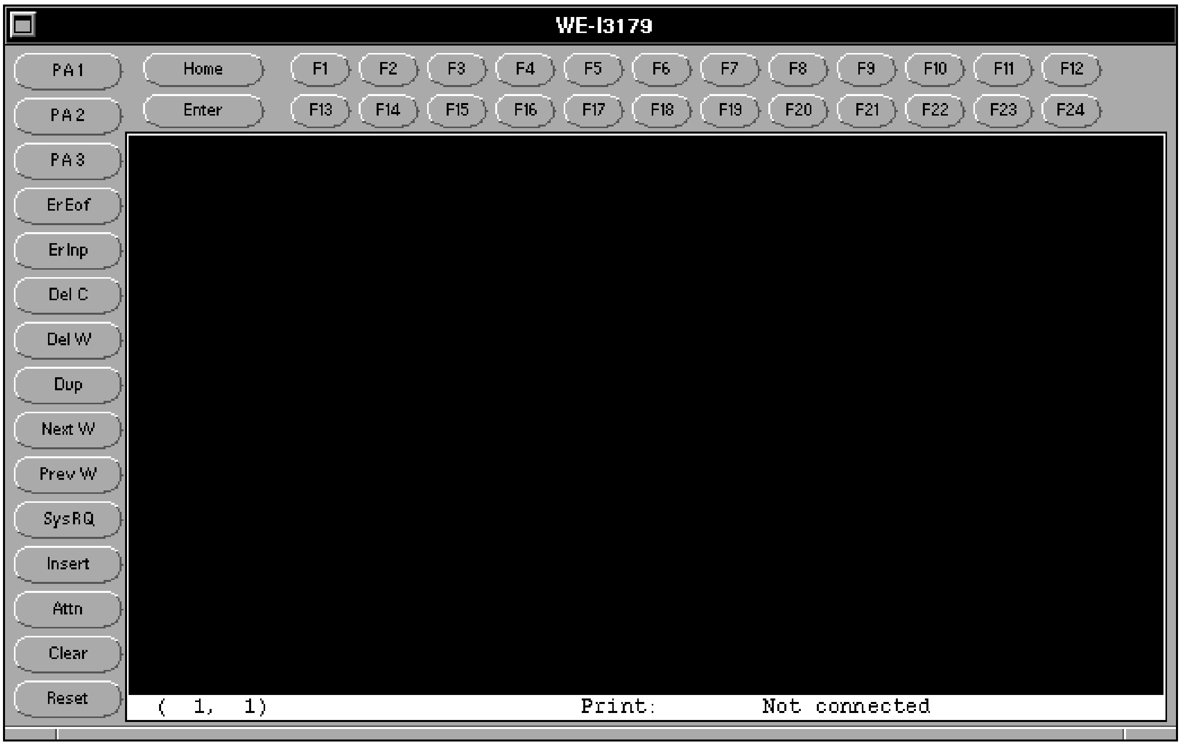
-> On X11 servers, the button width on top-right panel is determined by the length of their title. All other buttons (top-left panel and left panel) have a fixed width.

-> 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:



```
(  1,  1)                          Print:        Not connected
```

34. 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-I3179/disk_WE-I3179.dat

– Mandatory: No

– Example:

Disk_Print ~/we-i3179/MyPrints.$ext

Creates files with sequential numbering in the user’s home directory. See discussion in chapter 6.2.4 on page 31 for the meaning of $ext and how it will be replaced by WE-I3179.

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.

35. Print_Program *

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

– Defaults: lpr -r -h $FILE

– Mandatory: No

36. Print_Dest *

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

Printer
Disk

– Defaults: Printer

– Mandatory: No

37. 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

38. Print_Format *

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

Text only
Formatted
PostScript

– Defaults: Text only

– Mandatory: No

REMARK: There is currently no difference between Text only and Formatted.

39. Print_Output_Size *

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

Letter
A3
A4
A5

– Defaults: Letter

– Mandatory: No

40. 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

41. 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

42. Print_Generic_File

– Definition: Will define the name and path of the PostScript generic file which WE-I3179 will use as a template to generate PostScript:

~/we-i3179/ps_generic.dat

– Defaults: None

– Mandatory: Yes, if you want to generate PostScript and you don’t have a 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.

43. 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

NOTE: The password is different for “WE-I3179c” (character only emulation) and “WE-I3179g” (character and graphic emulation). This allows a user to begin with a character emulation and upgrate to the 3179g graphics emulation without reinstalling a new product. Only the password will be changed.

A.2 Fonts specifications

1. All fonts used with “WE-I3179” **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 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	8, 10, 12, 14, 18, 19, 24
Start_Height	12
Font_P	Ohlfs

B About colors and graphics

As mentioned at the beginning of this manual, the WE-I3179 terminal emulation may be purchased in two versions:

WE-I3179c	Emulates IBM 3270 character terminals models 2-5 with extended highlighting.
WE-I3179g	Same as WE-I3179c plus 3179g color graphics.

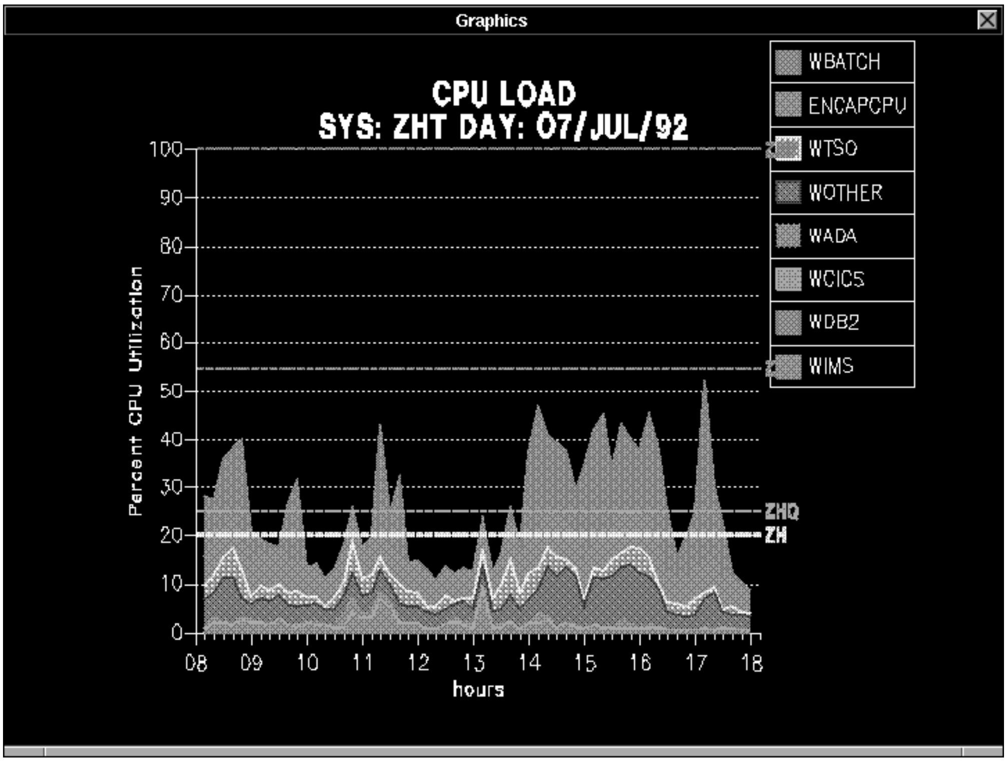
Although the configuration of WE-I3179 is quite simple forward, there are some points to remember if you want to display colors and graphics.

1. If you purchased WE-I3179c, you got a password which disallows 3179g graphics. However, the emulation will not crash if graphic orders are received. If this occurs, the graphic orders will merely be ignored instead of being displayed in a separate graphic window.

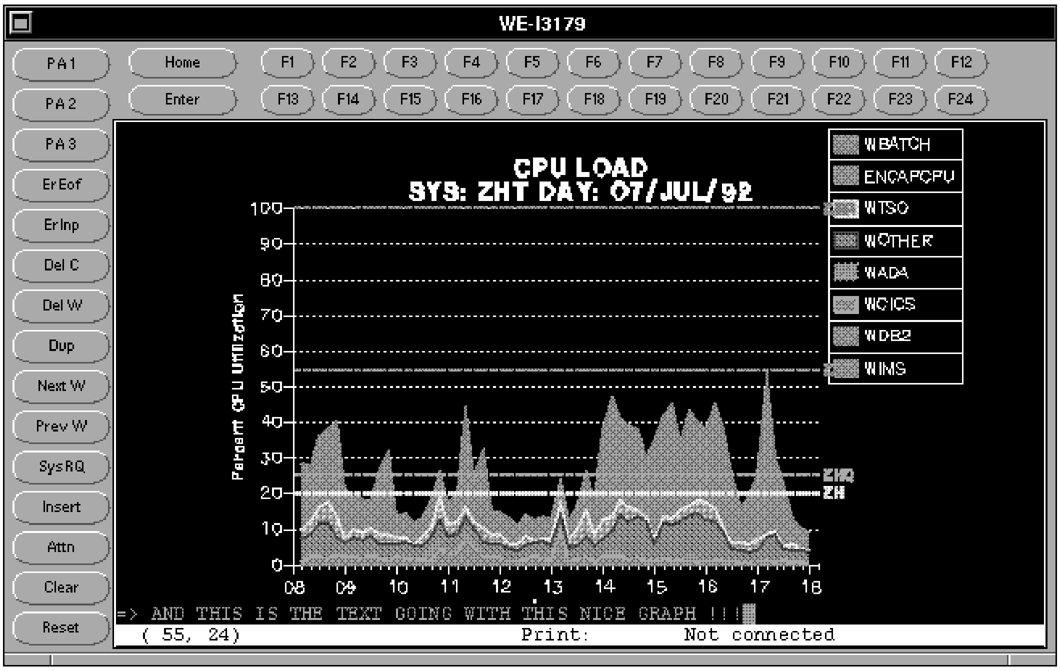
2. When a graphic screen is received, it will be displayed either in the same window as the text (as on an original 3179g terminal) or in a separate window created dynamically. This window can be resized, moved, hidden, etc. as any other window in your workspace. At start-up, the "`Graphic_Layer`" parameter (see on page 64) will decide whether graphics will be displayed in a separate window or merged with text. You can switch back and forth between merged and unmerged mode at any time (even while a graphic is displayed) by clicking the appropriate button in the parameter panel (Goodies menu).

3. If your graphic is displayed in a separate window, the last graphic screen received will remain in that window until a new graphic arrives or you close this window. This means that you can go on with your work within the character window without loosing your last graphical output. When you no longer need this last graphic, simply close this window. If new graphic arrives while the graphic window is closed, a new graphic window will be opened.

The illutration below shows a typical graphic screen output (separate graphic window).

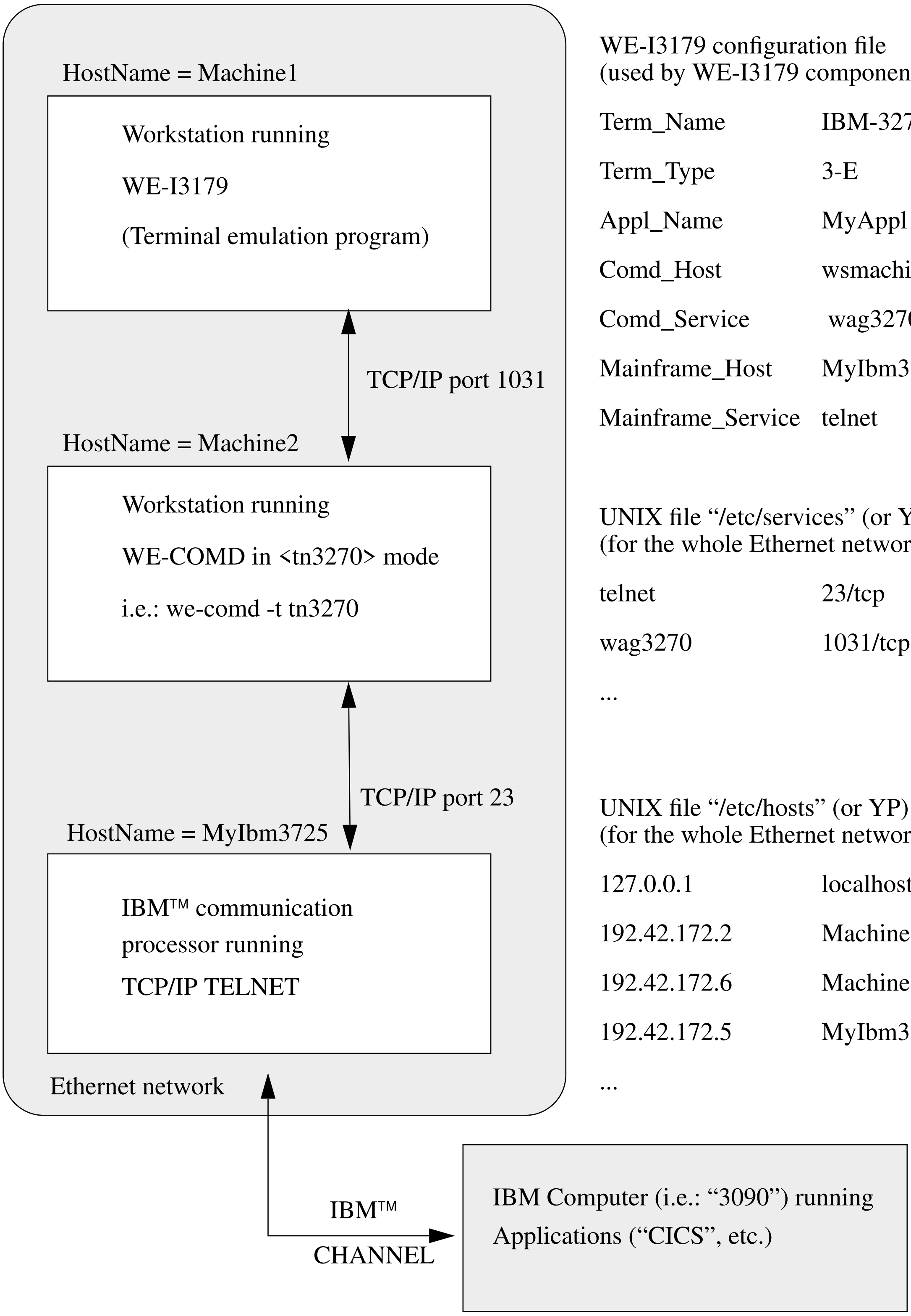


And below the same graphic merged with some text in the text window.



```
=> AND THIS IS THE TEXT GOING WITH THIS NICE GRAPH !!!
 ( 55, 24)                        Print:        Not connected
```

C TELNET example



WE-I3179 configuration file
(used by WE-I3179 component)

Term_Name	IBM-3278
Term_Type	3-E
Appl_Name	MyAppl
Comd_Host	wsmachine2
Comd_Service	wag3270
Mainframe_Host	MyIbm3725
Mainframe_Service	telnet

UNIX file "/etc/services" (or YP)
(for the whole Ethernet network)

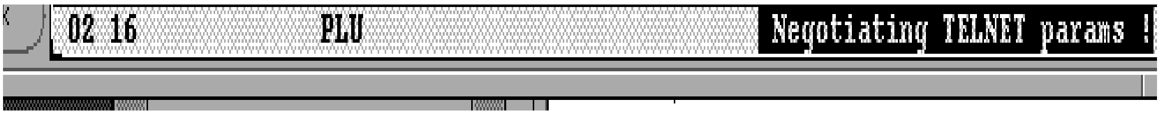
telnet	23/tcp
wag3270	1031/tcp
...	

UNIX file "/etc/hosts" (or YP)
(for the whole Ethernet network)

127.0.0.1	localhost
192.42.172.2	Machine1
192.42.172.6	Machine2
192.42.172.5	MyIbm3725
...	

D Status line, window border line, icon and messages

D.1 The status line fields



```
02 16          PLU                         Negotiating TELNET params !
```

Cursor coordinates PLU field Cmmunication status field

D.2 Cursor coordinates

This field indicates you the current cursor position in lines and columns.

D.3 PLU field

This field is not used for TELNET connections.

D.4 Normal messages displayed in the communication status field

D.4.1 Server: No connection

This is the normal message at start-up. It is mostly displayed so briefly that you may never notice this message.

D.4.2 => Negotiating TELNET params !

This is the normal message which is displayed while the <we-comd> server is attempting to handle the TELNET parameters with the IBM™ communication controller.

D.4.3 => 3270 session established

This is the normal message which must be displayed when you are connected to any host application. When this is the case, the upper window border contains a text like:

IBM-3278-2-E: Session <3090> on system <MyIbm3725>

Where:

- `IBM-3278` is your “Terminal_Name”.
- `2-E` is your “Terminal_Type”.
- `3090` is your “Appl_Name”.
- `MyIbm3725` is the Internet host name of the IBM communication controller you are connected to.

D.4.4 => Connect to HOST FAILED

This message is displayed at start-up if the connection to the host failed and WE-I3179 / WE-COMD have no better diagnostic to display.

D.4.5 => Connect CLOSED by HOST

This message is displayed when the IBM™ host closed the connection. You can restart the same connection by using the <Reconnect> entry of the Main menu.

REMARK: Under TELNET and depending on your host configuration, trying going back from the 1st screen displayed at start-up may produce a host disconnect.

D.4.6 Write ERROR to HOST => CLOSE

The last data couldn't be transmitted to the IBM™ host. If this message occurs, the communication to the IBM™ communication controller was interrupted and you could try a <Reconnect> operation from the Main menu. If this does not work and you get a message `(Socket to HOST FAILED!),` this probably indicates a malfunction of the IBM™ communication controller or more probably a server configuration error of the WE-I3179. Let a system administrator check the following:

- Proper functioning of the IBM™ communication controller (TCP/IP TELNET software running?).
- Proper entries in the UNIX "services" and "hosts" files.

D.4.7 ERROR, NO HOST connection

This message is displayed if you try a transmission (Enter or any PF or PA key) and the message `(3270 Session established)` is not displayed on your window status line. You should attempt a <Reconnect> from the Main menu to establish a new connection.

D.5 Configuration error messages displayed in the communication status field

Most configuration errors (i.e.: invalid fonts, missing parameters, etc.) will disallow WE-I3179 to start (**no** terminal window at all) and you will get an error message on the console or on your shell window. However, some configuration errors can only be detected when the first connection to the IBM™ host (through the communication processor) is attempted. One of the next three messages can then be displayed:

D.5.1 => CONFIG Err:Service NOT FOUND

Means that the "Mainframe_Service" entry in the WE-I3179 config file(s) have no counterpart in the UNIX "/etc/services" (or YP) file. You will have to correct that before retrying.

D.5.2 => CONFIG Err:Host NOT FOUND !

Means that the “Mainframe_Host” entry in the WE-I3179 config file(s) have no counterpart in the UNIX “/etc/hosts” (or YP) file. You will have to correct that before retrying.

D.5.3 => Socket to HOST FAILED !

Means that the “Mainframe_Service” or the “Mainframe_Host” entries in the WE-I3179 config file(s) have been found in “/etc/services” and “/etc/hosts” (or YP) but that a “socket” of the proper type couldn’t be created. Check with your UNIX system administrator.

D.6 Fatal error messages displayed in the communication status field

D.6.1 => WE-COMD EXIT WITH CODE x !!

If this message appears,some abnormal conditions occurred in the “WE-COMD” communication server. You will have to close this WE-I3179 window and reopen a new one to continue your work. This will also affect other WE-I3179 emulation connected to the same WE-COMD server.

E “M_func_names list”

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

E.1 Send AID's and data (if so defined by IBM) to the host

M_PF1	PF1 to host
M_PF2	PF2 to host
M_PF3	PF3 to host
M_PF4	PF4 to host
M_PF5	PF5 to host
M_PF6	PF6 to host
M_PF7	PF7 to host
M_PF8	PF8 to host
M_PF9	PF9 to host
M_PF10	PF10 to host
M_PF11	PF11 to host
M_PF12	PF12 to host
M_PF13	PF13 to host
M_PF14	PF14 to host
M_PF15	PF15 to host
M_PF16	PF16 to host
M_PF17	PF17 to host
M_PF18	PF18 to host
M_PF19	PF19 to host
M_PF20	PF20 to host
M_PF21	PF21 to host
M_PF22	PF22 to host
M_PF23	PF23 to host
M_PF24	PF24 to host
M_PA1	PA1 to host
M_PA2	PA2 to host
M_PA3	PA3 to host
M_ENTER	Enter to host
M_ATTN	Attention to host
M_CLEAR	Clear to host
M_SYSRQ	System request to host

E.2 Functions which may send data to the host (depending on macro content)

M_MACRO1EXEC	Execute macro 1
M_MACRO2EXEC	Execute macro 2
M_MACRO3EXEC	Execute macro 3
M_MACRO4EXEC	Execute macro 4
M_MACRO5EXEC	Execute macro 5

M_MACRO6EXEC	Execute macro 6
M_MACRO7EXEC	Execute macro 7
M_MACRO8EXEC	Execute macro 8
M_MACRO9EXEC	Execute macro 9
M_MACRO10EXEC	Execute macro 10
M_MACRO11EXEC	Execute macro 11
M_MACRO12EXEC	Execute macro 12

E.3 Functions executed locally by the emulator

M_EREOF	Erase from cursor to end of field
M_ERINP	Erase all input fields
M_TAB	Forward tab (go to next field if formatted)
M_BTAB	Backward tab (go to begin of previous field if formatted)
M_NEWLINE	New line (go to first field of next line if formatted)
M_CRLEFT	Move cursor 1 position to the left
M_CRRIGHT	Move cursor 1 position to the right
M_CRUP	Move cursor 1 position upward
M_CRDOWN	Move cursor 1 position downward
M_CRHOME	Move cursor to home position
M_INS_BACKSPACE	A **special** backspace function which will act like the backspace function while in insert mode
M_BACKSPACE	The normal (IBM defined) backspace function
M_DELC	Delete character
M_DELW	Delete word
M_DUP	DUP function
M_FMARK	Field mark function
M_INSERT	Set insert mode
M_RESET	Execute reset function (Reset key lock and insert mode)
M_PRINT	Execute hardcopy of text screen
M_UNLOCK	Unlock keyboard
M_COPY	Copy current selection to pasteboard
M_PASTE	Paste current pasteboard content at current cursor location

Index