



Digi International Inc.
6400 Flying Cloud Drive
Eden Prairie, MN 55344
(800) 344-4273
(612) 943-9020

Digi International GmbH
Domkloster 1
50667 Köln
Germany
+49 221 920520

Digi International Asia Pte Ltd
13-06 Tower 'A'
391A Orchard Road
Ngee Ann City
Singapore 0923
+65 732 1318

Software Manual

**RealPort Device Driver
for
SCO UNIX System V Release 3.2
SCO OpenServer
UNIX System V Release 4**

92000159D

DigiWARE™, **DigiPRINT™**, **DigiSCREEN™**, **RealPort™**, **PortServer™**, **PortServer II™** and the Digi logo are trademarks of Digi International Inc. All other brand and product names are the trademarks of their respective holders.

© Digi International Inc. 1994, 1995
All Rights Reserved

Digi International Inc. 6400 Flying Cloud Drive Eden Prairie, MN 55344	Phone (800) 344-4273 (612) 943-9020 FAX (612) 943-5398 BBS (612) 943-0550
---	--

Digi International GmbH Domkloster 1 50667 Köln Germany	Phone +49 221 920520 FAX +49 221 9205210 BBS +49 221 9205211
---	--

Digi International Asia Pte Ltd 13-06 Tower 'A' 391A Orchard Road Ngee Ann City Singapore 0923	Phone +65 732 1318 FAX +65 732 1312 BBS +65 292 7163
---	--

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is”, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Table of Contents

Introduction	5
Software Installation — SCO.....	6
Linking a New Kernel.....	6
Creating New Devices.....	12
Installing the New Kernel	13
Enabling the New Ports	14
Serial Printers	16
Un-Installing the SCO UNIX Device Driver	17
Digi TTY Devices (SCO UNIX)	19
Software Installation — SVR4.....	21
Installing the Driver Package.....	21
Creating New Devices.....	27
Installing the New Kernel	28
Enabling the New Ports	29
Serial Printers	31
Un-Installing the RealPort Device Driver.....	32
Digi TTY Devices (SVR4).....	33
Setting Terminal Options with ditty.....	35
Using the ditty Command with Parallel Ports.....	39
DigiPRINT Transparent Print Feature	41
Description & Theory of Operation	41
Transparent Print Activation.....	43
DigiSCREEN Multiple Screen Utility	45
Description and Theory of Operation	45
Options.....	46
Using DigiSCREEN.....	47
Dynamic Screen Assignment	48
DSINFO DigiSCREEN Information File.....	49
Description	49
Entry Format	49
Error Messages.....	55

Digi Support Services	56
The Digi Bulletin Board System.....	56
Internet FTP Server	57
World Wide Web Server	57
DigiFACTs FaxBack Server.....	57
Technical Support.....	59
Customer Service.....	59
Return Procedures.....	60

Introduction

The Digi RealPort device drivers for SCO UNIX, SCO OpenServer and System V, Release 4 are installable device drivers; the software development system is *not* required to create a new kernel. The link kit, however, must be installed on SCO systems.

This manual also contains instructions for the following features:

- `ditty`, a utility program that sets and displays the terminal options for Digi intelligent serial products (page 35)
- DigiPRINT transparent printing (page 41)
- DigiSCREEN, Digi's multiple screen utility (page 45)

Once you have completed the hardware installation instructions in the PortServer *Installation Guide*, you may proceed with the software device driver installation instructions starting on the following page.



Except where noted, *PortServer* refers to both PortServer and PortServer II.



Be sure to read the *Release Notes* that may be included with this software device driver. The *Release Notes* contain information not available at this manual's press time.

Software Installation — SCO



The TCP/IP Runtime System must be installed for the PortServer and this device driver to work.



Software changes more rapidly than printed documentation can keep up. For this reason, some of the screens or prompts may not appear exactly as shown.

Linking a New Kernel

1. Log onto the console as super-user (root).
2. Insert the DigiWARE diskette, and enter:

```
custom
```
3. When the operating system menu asks, choose **INSTALL**, then select **A NEW PRODUCT**. Note that if you are *re-installing* a previously removed device driver, you should still choose New Product at this point.
4. When prompted again, select **ENTIRE PRODUCT**. When asked to insert the distribution disk, insert the Digi-supplied device driver software disk for SCO UNIX.
5. You will see the following screen:

```
Copyright (c) 1994-95 Digi International Inc.      All Rights Reserved
RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government
is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the
Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.  Digi
International Inc., 6400 Flying Cloud Dr., Eden Prairie, MN 55344.
If this is unacceptable, press <Q> to quit, <CR> to proceed:
```

6. Press <Enter>. You will see:

```
This script installs the RealPort driver for:

      SCO OpenServer Release 5
      SCO UNIX System V/386 Release 3.2

This script also installs the information needed by UNIX to use
these additional ports.

By default, the ports are installed with the basename "tty".
DigiPRINT ports are installed with the basename "pr".

      Is this acceptable (y/n)?
```

If you answer “n”, you will be prompted for new basenames.

7. You will now see:

```
The PortServer installation will require that you provide
the IP address and the TCP port number of the PortServer.

Alternatively you may provide the PortServer's name, as known
to the nameserver and the TCP port number of the PortServer.

Do you wish to continue at this time (y/n)?
```

8. Answer “y”. The system displays:

```
The port names (in the /dev directory) will look like:

      /dev/ttya01

where `tty' is the basename you picked above, `a' is an ID letter
that can be different for each port box, and `01' is the port number.

Systems with a large number of ports may need to use some letters more
than once. When this occurs, the first box to use an ID letter will
get port numbers 01-16, the second box to use that letter will get
port numbers 17-32, and so on.

You can have the install script pick ID letters for you, assigning
letters consecutively (a,b,c,d,...) and starting over from the
beginning when it runs out of letters; or you can choose them manually.
Manual ID selection is recommended for larger systems, where it can be
used to help keep port names more organized.

      Do you want ID letters picked for you (y/n)?
```

If you answer “y”, ID letters will be automatically picked. If you answer “n”, you will be asked (later in the procedure) to specify an ID letter for each PortServer.

9. Now the system will ask:

```
How many PortServers do you wish to install (1-64)?
```

Enter the number of PortServer modules you have connected to your network.

10. You will now see:

```
Configuring server 1.  
Do you wish to use the  
    (I)P address  
or the  
    (N)ame known to the name server  
for this PortServer (I/N)?
```

If you enter “I”, you will be prompted for the IP address of the PortServer. Enter the appropriate IP address (see the PortServer *Installation Guide*).

If you enter “N”, you will be asked for the PortServer’s name. This name must be registered in `/etc/hosts` or a name server.

11. Now you will be asked for the RealPort TCP port number:

```
Enter the RealPort TCP port number (default 771):
```

Enter the appropriate TCP port number (the RealPort default is 771).



If you use a TCP port number that is different from the default, be sure to set the PortServer for the same number. This is done with the `set config` command (see the Command Reference section of the PortServer *Installation Guide*).

12. Next, you will be asked for the number of ports on the first PortServer module:

```
Configuring server 1.  
How many ports are on server 1 (default 16):
```

Enter the total number of physical ports on the PortServer. If you have a PortServer II, include the ports on any attached PORTS modules (PORTS/16em, PORTS/8em, PORTS/8emp). Be sure to include the parallel printer port if the PORTS module is a PORTS/8emp.

13. If you answered “n” to the question “Do you want ID letters picked for you?” back in Step 8, you will also be prompted at this time for the ID letters to be assigned to the PortServer.

14. You will now be shown the configuration you have selected:

```
You have selected the following configuration for server 1:  
  
IP address:      192  
TCP port:       771  
Port Names:     ttya01 - ttya16  
  
Is this configuration acceptable (y/n)?
```

If you answer “n”, you will be returned to Step 10.

Steps 10-14 will be repeated for each PortServer module specified in Step 9.

15. Now you will be asked:

```
Do you wish to install DigiSCREEN (y/n)?
```

DigiSCREEN is a multiple screen utility that allows a single physical terminal to be connected to several virtual terminal sessions (*screens*) at one time. See the discussion of DigiSCREEN beginning on page 45.

If you have already installed DigiSCREEN with another Digi product, there is no need to install it again, unless you wish to change the number of DigiSCREEN devices.

16. If you answered “y” in step 16, you will be asked:

```
Each session under DigiSCREEN requires a DigiSCREEN device.  
How many DigiSCREEN devices do you wish to have installed?  
(max. is 256, default is 32)
```

Answer with the number of DigiSCREEN devices you think will be adequate for all connected terminals. Note that the default value will change according to the number of previously created DigiSCREEN devices you may have. See page 45 for a complete description of the DigiSCREEN multiple screen utility.

17. Next, the software asks:

```
Do you wish to update /etc/ttytype (y/n)?
```

For a new installation, answer “y” (yes). For updates, answer “n” to preserve the current setup. If you are not sure, answer “y”.

18. Finally, you are asked:

```
Do you wish to copy the software version of the PortServer firmware to your host at  
this time (y or n)?
```

Answer “y”.

The PortServer’s internal operating software can be upgraded without ROM or other hardware changes. Refer to the Command Reference section of the PortServer *Installation Guide* for information on setting up the PortServer for remote booting.

Important

The boot filename and the IP address of the boot host must be registered in the PortServer's internal configuration to enable remote booting. This is done with the `set config` command (see the Command Reference section of the PortServer *Installation Guide*). The boot host must be running `tftp`.

Example:

```
set config boothost ip_address
set config bootfile Boot_File
```

When you answer “y” to the above question, a file name will be provided. Use this name for *Boot_File*.

If *Boot_File* is not in the directory `/tftpboot`, the full pathname of the file must be included in the `set config` command.

Creating New Devices

This portion of the installation procedure creates devices in `/dev`, giving UNIX the information to use the additional ports.

The devices are named according to the following conventions:

Format used for port names

- tty** Default base name. You may have selected a different one.
- pr** Default base name for DigiPRINT transparent printing. You may have selected a different one.
- a-z** PortServer letter I.D. Legal letters are a-z; ports (tty or printer) with modem control are identified by a capital letter A-Z.
- 01-96** Port number on the individual PortServer.

For example, if “**tty**” is the base name, Port 4 on PortServer “**b**” will be **ttyb04**. The same port with modem control enabled would be named **ttyB04**.

Installing the New Kernel

The device driver software installation is now complete, and the new kernel must be installed.

1. You will see:

```
Do you want this kernel to boot by default (y or n)?
```

Answer “y” (yes). The system then displays:

```
Do you want the kernel environment rebuilt (y or n)?
```

Answer “y” again. When you are returned to the custom menu, select “Q” to quit.

2. Enter the following command to boot the new kernel:

```
shutdown -y -g0
```

Enabling the New Ports

1. Connect terminals to the ports (using a null modem, if necessary) and test the connections to each terminal by entering the following command for *each* port added:

```
date > /dev/ttya01
```

(Assuming the terminal is connected to `ttya01`.)

(Please note that in the above, the “date” command is used as a simple test, to provide text output that can be redirected; there is no other significance to “date” in this test.)

- If the date appears on the terminal’s screen, the device is properly connected.
- If the date *does not* appear on the terminal’s screen, then that terminal is not receiving data; check the power, cables, connections, etc.
- If nonsense characters are printed on the terminal’s screen, check the baud rates, data bits, stop bits, and parity setting on your terminal (UNIX default parameters are 9600 baud, 8 data bits, 1 stop bit, and no parity).

Another test you can perform will verify that a port can both transmit and receive data:

Connect the port’s transmit and receive lines together (pins 2 and 3 of a DB-25 connector, or the middle two pins of an RJ-45 connector), and enter the following commands:

```
cat </dev/ttya01 &  
cat /etc/termcap > /dev/ttya01
```

The first command runs in the background, and directs all input from `ttya01` to `stdout` (your screen). The second command transmits the `termcap` file to `ttya01`. If the port is working, the data is sent out the port, received back by the same port and displayed on your screen.

If the port fails, check another port. If more than one port fails, there is probably a driver installation problem (TCP port numbers don’t agree, IP address is invalid, or some other networking problem).

Once you can redirect output to a terminal with the test above, perform the following steps to enable that port for login.

Important!

If the port tests above work, but you still can't respawn a `getty`, it is probably because DCD is not asserted.

Remember that modem controlled ports (where the PortServer ID is a capital letter—e.g. `ttyA01`) must have Data Carrier Detect (DCD) asserted before they can function. If DCD is not connected to an active output line of the peripheral device, it can be defeated by “tying it high” (e.g. connecting it to the same port's DTR line).

Another way to bypass the DCD requirement is to use non modem control devices (where the PortServer ID is a small letter—e.g. `ttya01`). Non modem control devices function identically to modem control devices, except that they do not require DCD in order to function.

2. The Terminal Control Database must be updated to include the new devices. Add the Digi device information using the following SCO UNIX `sysadmsh` command sequence:

Accounts→Terminal:Create

Fill in the required information on the form displayed.

3. To activate a port for use with a terminal, enter the following command:

```
enable ttya01
```

The above command will cause a login prompt to be sent to the terminal connected to **ttya01**. To activate the other ports, repeat the `enable` command with the appropriate port name.

Serial Printers

To prepare a port for connection to a printer, enter the following command:

```
sleep 999999 < /dev/ttya01 &
```

This command forces the specified port to stay open, thus preserving any “non-sticky” options (see *Setting Terminal Options with ditty*, beginning on page 35, for complete descriptions of all `ditty` parameters) for that port.

This command is recommended for any port that has a printer connected to it.

The command may be placed in `/etc/rc.d/8/userdef` so that it is automatically executed whenever the system is booted.

Un-Installing the SCO UNIX Device Driver

The Digi device driver software can be *removed* from the kernel by using the program `custom`. Again, the Development Kit is not required, although the link kit must be installed.

Removing the driver from the kernel is a two-part process. First, the driver references are removed from the system configuration files. Secondly, a new kernel is linked, and copied to the root directory.

Enter the following commands to *un-install* the device driver software:

1. Log onto the console as super-user (root), and enter the following command:

```
custom
```

2. When the operating system menu asks, choose REMOVE.

3. Select **DigiWARE for PortServer**.

4. Select the Digi driver. The software will respond with:

```
Removing Digi PortServer driver.
```

5. Next, the software asks:

```
Do you wish to remove PortServer entries from /etc/ttytype (y or n)?
```

Answer “y” if these entries are no longer needed (or if you are in doubt), or answer “n” if you wish to preserve the old entries.

6. You will see:

```
Do you want this kernel to boot by default (y or n)?
```

Answer “y”. The system displays:

```
Do you want the kernel environment rebuilt (y or n)?
```

Answer “y”. You will see:

```
The kernel has been successfully linked and installed. To activate it, reboot your system.
```

7. Enter the following command to boot the new kernel:

```
shutdown -y -g0
```

Digi TTY Devices (SCO UNIX)

This device driver supports three different UNIX devices on each line. On line “a01,” where “a” refers to PortServer **a**, and “01” refers to first line (port) on that concentrator, there are four devices:

`/dev/ttya01`

Standard device. This is the same as `/dev/ttyA01` (see below), with the exceptions that the default handshake method is XON/XOFF, and that Data Carrier Detect need not be present to open the device. *Once a connection is established and DCD becomes active, standard devices behave in the same way as modem devices—subsequent loss of the Data Carrier Detect signal will cause the jobs to be killed and the user will be automatically logged off.*

`/dev/ttyA01`

The “modem” tty device, used for terminals, modems, printers, laboratory equipment, etc. This is also sometimes referred to as a “dial-in” device.

This device is a traditional UNIX port with modem control. It requires Carrier Detect to be high before it will operate.

When used with a modem, the port will wait for carrier before sending out the `login:` prompt, so the user is greeted properly upon making a connection.

When used with a terminal or other device, it is usually wise to wire the Digi DCD signal to the terminal’s DTR (Data Terminal Ready) line. When the terminal is turned on, the system outputs a `login:` prompt. When the terminal is turned off, any associated jobs are killed, and the user is logged out.

```
/dev/pra01  
/dev/prA01
```

The “Transparent Print Devices” (DigiPRINT) for use with the auxiliary printer port of a terminal. Output directed to a `pr` device goes out the auxiliary port of a terminal while the user continues to use the terminal normally.

Use `pra01` if your terminal device is `ttya01` (standard device); use `prA01` if your terminal device is `ttyA01` (modem device).

Transparent print devices allow you to use your terminal and a local printer connected to the terminal’s auxiliary port at the same time. The system assumes data you send to the `pr` device is destined for the printer. Before sending data to the printer, the system sends a special control sequence to the terminal to activate the printer port, sends the data, then turns the printer port back off again before sending more data for the terminal.

For correct operation, certain parameters must be provided to the *ditty* program. It is usually best to set these up in `/etc/rc2.d`, since they must be set each time the system is rebooted. See *Setting Terminal Options with ditty* (page 35) and *DigiPRINT Transparent Print Option* (page 41).

An example setup line in one of the `/etc/rc` files might be:

```
ditty term wyse60 maxcps 80 bufsize 2000 maxchar 50 /dev/ttya01
```

Software Installation — SVR4



Software changes more rapidly than printed documentation can keep up. For this reason, some of the screens or prompts may not appear exactly as shown.

Installing the Driver Package

1. Log onto the console as super-user (root).
2. To begin the installation, insert the DigiWARE diskette in the diskette drive and enter the command:

```
pkgadd -d diskette1
```

(or pkgadd -d diskette2 if the diskette is in the second drive)

This will run the System Administration program designed to assist in Add On Package installation.

Refer to the Operations/System Administrator's Guide for more information on `pkgadd`.

You will see the following (or something similar):

```
Installation in progress. Do not remove the diskette.

The following packages are available:

1.  ncxa    Digi Async Driver Package V. x.x.x

Select package(s) you wish to process (or 'all' to process all packages). (default:
all) [?, ??, q]
```

3. Press <Enter> to select the RealPort device driver for PortServer.

4. You will see the following screen:

```
Copyright (c) 1994-95 Digi International Inc.    All Rights Reserved

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government
is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the
Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.  Digi
International Inc., 6400 Flying Cloud Dr., Eden Prairie, MN 55344.

If this is unacceptable, press <Q> to quit, <CR> to proceed:
```

5. Press <Enter>. You will see:

```
This script installs the RealPort driver for:

    UNIX System V Release 4.x

This script also installs the information needed by UNIX to use
these additional ports.

By default, the ports are installed with the basename "term/".
DigiPRINT ports are installed with the basename "pr".

    Is this acceptable (y/n)?
```

If you answer “n”, you will be prompted for new basenames.

6. You will now see:

```
The PortServer installation will require that you provide
the IP address and the TCP port number of the PortServer.

Alternatively you may provide the PortServer's name, as known
to the nameserver and the TCP port number of the PortServer.

Do you wish to continue at this time (y/n)?
```

7. Answer “y”. The system displays:

```
The port names (in the /dev directory) will look like:

    /dev/term/a01[ms]

where `term/' is the basename you picked above, `a' is an ID letter
that can be different for each port box, and `01' is the port number.

Systems with a large number of ports may need to use some letters more
than once. When this occurs, the first box to use an ID letter will
get port numbers 01-16, the second box to use that letter will get
port numbers 17-32, and so on.

You can have the install script pick ID letters for you, assigning
letters consecutively (a,b,c,d,...) and starting over from the
beginning when it runs out of letters; or you can choose them manually.
Manual ID selection is recommended for larger systems, where it can be
used to help keep port names more organized.

Do you want ID letters picked for you (y/n)?
```

If you answer “y”, ID letters will be automatically picked. If you answer “n”, you will be asked (later in the procedure) to specify an ID letter for each PortServer.

8. Now the system will ask:

```
How many PortServers do you wish to install (1-64)?
```

Enter the number of PortServer modules you have connected to your network.

9. You will now see:

```
Configuring server 1.

Do you wish to use the
    (I)P address
or the
    (N)ame known to the name server
for this PortServer (I/N)?
```

If you enter “I”, you will be prompted for the IP address of the PortServer. Enter the appropriate IP address (see the PortServer *Installation Guide*).

If you enter “N”, you will be asked for the PortServer’s name. This name must be registered in `/etc/hosts` or a name server.

10. Now you will be asked for the RealPort TCP port number:

```
Enter the TCP port number for this PortServer:
```

Enter the appropriate TCP port number (the RealPort default is 771).



If you use a TCP port number that is different from the default, be sure to set the PortServer for the same number. This is done with the `set config` command (see the Command Reference section of the PortServer *Installation Guide*).

11. Next, you will be asked for the number of ports on the first PortServer module:

```
How many ports are on PortServer 1 (default 16):
```

Enter the total number of physical ports on the PortServer. If you have a PortServer II, include the ports on any attached PORTS modules (PORTS/16em, PORTS/8em, PORTS/8emp). Be sure to include the parallel printer port if the PORTS module is a PORTS/8emp.

12. If you answered “n” to the question “Do you want ID letters picked for you?” back in Step 7, you will also be prompted at this time for the ID letters to be assigned to the PortServer.

13. Finally, you will be shown the configuration you have selected:

```
You have selected the following configuration for server 1:  
  
IP address:      192.192.192.192  
  
TCP port:       771  
  
Port Names:     term/g01[ms] - term/g16[ms]  
  
Is this configuration acceptable (y or n)?
```

If you answer n, you will be returned to Step 9.

Steps 9-13 will be repeated for each PortServer module specified in Step 8.

14. Finally, you are asked:

```
Do you wish to copy the software version of the PortServer firmware to your host at
this time (y or n)?
```

Answer “y”.

The PortServer’s internal operating software can be upgraded without ROM or other hardware changes. Refer to the Command Reference section of the PortServer *Installation Guide* for information on setting up the PortServer for remote booting.

Important!

The boot filename and the IP address of the boot host must be registered in the PortServer’s internal configuration to enable remote booting. This is done with the `set config` command (see the Command Reference section of the PortServer *Installation Guide*). The boot host must be running `tftp`.

Example:

```
set config boothost ip_address
set config bootfile Boot_File
```

When you answer “y” to the above question, a file name will be provided. Use this name for *Boot_File*.

If *Boot_File* is not in the directory `/tftpboot`, the full pathname of the file must be included in the `set config` command.

Creating New Devices

Devices will now be created in `/dev`, giving the system the information to use the additional ports.

The devices are named according to the following conventions:

Format used for port names

`/dev/term/`

Directory path for tty devices.

`pr` Default base name for DigiPRINT transparent printing. Transparent printer devices are placed in `/dev`.

`a-z` PortServer letter I.D. Legal letters are a-z.

`01-96` Port number on the individual PortServer.

`s, m` Indicates whether port is a “standard” (**s**) or “modem” (**m**) port. Modem ports can only be opened when Data Carrier Detect (DCD) is active.

For example, standard tty port 4 on concentrator or PortServer “**b**” will be `/dev/term/b04s`. The same port with modem control enabled would be named `/dev/term/b04m`

NOTE: A port may be enabled as either a standard or modem tty device, but not both.

You may see the following message during device creation:

```
This system's MAXMINOR value is not sufficient for the number of
devices installed at this time.
```

```
Please increase that value in /etc/conf/mtune.d/io prior to
running /etc/conf/bin/idtune MAXMINOR 64
Hit <CR> to proceed:
```

If the above message is displayed, refer to the System Administrator’s manual for the procedure to increase the MAXMINOR parameter. It must be a value at least four times as large as the number of ports installed.

Installing the New Kernel

The device driver software installation is now complete, and the new kernel must be installed.

You will see:

```
Installing UNIX device information ... Done.  
Installing nxa driver.           Done.  
Installing RealPort driver.     Done
```

Enabling the New Ports

1. Connect terminals to the ports (using a null modem, if necessary) and test the connections to each terminal by entering the following command for *each* port added:

```
date > /dev/term/a01s
```

(Assuming the terminal is connected to a01s.)

Please note that in the above, the “date” command is used as a simple test, to provide text output that can be redirected; there is no other significance to “date” in this test.

Also note that the default settings for the driver are 9600 baud, 8 data bits, no parity and 1 stop bit.

- If the date appears on the terminal’s screen, the device is properly connected.
- If the date *does not* appear on the terminal’s screen, then that terminal is not receiving data; check the power, cables, connections, etc.
- If nonsense characters are printed on the terminal’s screen, check the baud rates, data bits, stop bits, and parity setting on your terminal.

Another test you can perform will verify that a port can both transmit and receive data:

Connect the port’s transmit and receive lines together (pins 2 and 3 of a DB-25 connector, or the middle two pins of an RJ-45 connector), and enter the following commands:

```
cat </dev/term/a01s &  
cat /etc/termcap > /dev/term/a01s
```

The first command runs in the background, and directs all input from a01s to stdout (your screen). The second command transmits the termcap file to a01s. If the port is working, the data is sent out the port, received back by the same port and displayed on your screen.

If the port fails, check another port. If more than one port fails, there is probably a driver installation problem (TCP port numbers don’t agree, IP address is invalid, or some other networking problem).

Once you can redirect output to a terminal with the test above, perform the following steps to enable that port for login.

2. Log onto the console as super-user (root).

3. Enter:

```
sysadm
```

This will bring up the **System Administration menu**. From this menu, select **“Ports”**. The **Ports** menu will now be displayed. From the **Ports** menu, select **“Quick Terminal”**. Now select **“Add”**.

You will now be shown a list of port numbers that are currently being monitored for activity (the list may be empty at this time).

3. Press <F2> (**“Choices”**). You will be shown a list of available ports. Use the arrow keys to move through the list of ports. Press <F2> to mark ports to be enabled (an asterisk will appear by the marked choices).

IMPORTANT! Do not select both modem and standard devices for the same port number (e.g. /dev/term/a01sand /dev/term/a01n).

4. When you have completed your selections, press <F3> (**“Enter”**). The list you saw in Step 3 will now include the ports you just selected. To accept the list, press <F3> (**“Save”**).

5. Now press <F3> again (**“Continue”**), press <F7> (**“Cancel”**) to exit the **Ports Setup** menu. The ports are now enabled.



The default settings for login ports are typically: **9600, ixon, cs7, e, 1** (these defaults may have been changed by your system administrator). These settings can be modified by going again to the **Ports** menu, selecting **Ports Services**, then selecting **Modify**.

Refer to your UNIX System Administrator's Guide, under the *Peripherals Setup* section, for more details on how to set up serial ports.

Serial Printers

To prepare a port for connection to a printer, enter the following command:

```
sleep 999999 < /dev/term/a01s &
```

This command forces the specified port to stay open, thus preserving any “non-sticky” options (see *Setting Terminal Options with ditty*, beginning on page 35, for complete descriptions of all `ditty` parameters) for that port.

This command is recommended for any port that has a printer connected to it.

The command may be placed in `/etc/rc2.d/S99RealPort` so that it is automatically executed whenever the system is booted (`S99RealPort` is an arbitrary file name; you may choose another).

Un-Installing the RealPort Device Driver

Enter the following commands to *un-install* the device driver software:

1. Log onto the console as super-user (root).
2. Enter the following command:

```
pkgrm -n ncxa
```

It will take some time for the new kernel to be built. When it is finished, you will be asked if you want the port monitor entries removed. If you answer “n” (no), they will be left intact (this is useful if you are installing a new revision of the device driver, and want the same ports enabled).

Now perform a system shutdown and reboot to activate the new kernel.

Digi TTY Devices (SVR4)

This device driver supports three different UNIX devices on each line. On line “a01,” where “a” refers to PortServer **a**, and “01” refers to first line (port) on that concentrator, there are four devices:

`/dev/term/a01m`

The “modem” tty device, used for terminals, modems, printers, laboratory equipment, etc. This is also sometimes referred to as a “dial-in” device.

The **XXm** device is a traditional UNIX port with modem control. It requires Carrier Detect to be high before it will operate. RTS/CTS handshaking is enabled by default.

When used with a modem, the port will wait for carrier before sending out the `login:` prompt, so the user is greeted properly upon making a connection.

When used with a terminal or other device, it is usually wise to wire the Digi DCD signal to the terminal’s DTR (Data Terminal Ready) line. When the terminal is turned on, the system outputs a `login:` prompt. When the terminal is turned off, any associated jobs are killed, and the user is logged out.

`/dev/term/a01s`

Standard device. This is the same as `/dev/term/a01m` with the exceptions that the default handshake method is XON/XOFF, and that Data Carrier Detect need not be present to open the device. *Once a connection is established and DCD becomes active, standard devices behave in the same way as modem devices—subsequent loss of the Data Carrier Detect signal will cause the jobs to be killed and the user will be automatically logged off.*

```
/dev/pra01s  
/dev/pra01m
```

The “Transparent Print Devices” (DigiPRINT) for use with the auxiliary printer port of a terminal. Output directed to a `pr` device goes out the auxiliary port of a terminal while the user continues to use the terminal normally.

Use `pra01s` if your terminal device is `term/a01s` (standard device); use `pra01m` if your terminal device is `term/a01m` (modem device).

Transparent print devices allow you to use your terminal and a local printer connected to the terminal’s auxiliary port at the same time. The system assumes data you send to the `pr` device is destined for the printer. Before sending data to the printer, the system sends a special control sequence to the terminal to activate the printer port, sends the data, then turns the printer port back off again before sending more data for the terminal.

For correct operation, certain parameters must be provided to the *ditty* program. It is usually best to set these up in `/etc/rc2.d`, since they must be set each time the system is rebooted. See *Setting Terminal Options with ditty* (page 35) and *DigiPRINT Transparent Print Option* (page 41).

An example setup line in one of the `/etc/rc` files might be:

```
ditty term wyse60 maxcps 80 bufsize 2000 maxchar 50 /dev/term/a01s
```

Setting Terminal Options with ditty

`ditty` is a utility program that sets and displays the terminal options for the Digi PortServer modules.

The format is:

```
ditty[-a] [-p] [-n ttyname] [option(s)] [ttyname]
```

With no options, `ditty` displays all Digi special driver settings, modem signals, and all standard parameters displayed by `stty(1)` for the tty device referenced by standard input.

Command options are provided to change flow control settings, set transparent print options, force modem control lines, and display all tty settings. Any unrecognized options are passed to `stty(1)` for interpretation.

The options are:

- `-a` Display all of the unique Digi option settings, as well as all of the standard tty settings reported by `stty -a`.
- `-p` Interpret the port as a parallel printer port. This flag should only be used for the ninth port of a PORTS/8emp module.
- `-n ttyname` Set and display options for the given tty device, instead of standard input. This option may be specified multiple times to perform the same operation on multiple ttys.
- ttyname* Set and display options for the given tty device, instead of standard input. This form can be used with a tty pathname prefixed by `/dev/` or with a simple tty name beginning with `tty` (or the basename you selected during installation, if different). This option may be used on a modem control line when no carrier is present.

The following options specify transient actions to be performed immediately:

- `break` Send a 250 MS break signal out on the tty line.
- `flush` Immediately flush (discard) tty input and output.
- `flushin` Flush tty input only.
- `flushout` Flush tty output only.

The following options specify actions which are not “sticky,” meaning that the changes are reset when the device is closed, and that the device will use the default values the next time it is opened.

<code>stopout</code>	Stop output exactly as if an xoff character was received.
<code>startout</code>	Restart stopped output exactly as if an xon character was received.
<code>stopin</code>	Activate flow control to stop input.
<code>startin</code>	Release flow control to resume stopped input.
<code>[-]dtr</code>	Raise [drop] the DTR modem control line, unless DTR hardware flow control is selected.
<code>[-]rts</code>	Raise [drop] the RTS modem control line, unless RTS hardware flow control is selected.

The following options are “sticky”—the effects continue until the system is rebooted or until the options are changed.

<code>[-]fastcook</code>	Perform cooked output processing on the intelligent card to reduce host CPU usage, and increase raw mode input performance.
<code>[-]fastbaud</code>	Alter the baud rate tables, so 50 baud becomes 57,600 baud, 75 baud becomes 76,800 baud and 110 baud becomes 115,200 baud.
<code>[-]rtspace</code>	Enable [disable] RTS hardware input flow control, so RTS drops to pause remote transmission.
<code>[-]ctspace</code>	Enable [disable] CTS hardware output flow control, so local transmission pauses when CTS drops.
<code>[-]dsrpace</code>	Enable [disable] DSR hardware output flow control, so local transmission pauses when DSR drops.
<code>[-]dcdpace</code>	Enable [disable] DCD hardware output flow control, so local transmission pauses when DCD drops.
<code>[-]dtrpace</code>	Enable [disable] DTR hardware input flow control, so DTR drops to pause remote transmission.
<code>[-]forcedcd</code>	Disable [re-enable] carrier sense, so the tty may be opened and used even when carrier is not present.

<code>[-]altpin</code>	No effect in this device driver. To enable alternate wiring for DCD and DSR, use the PortServer's <code>set flow</code> command. See the PortServer <i>Installation Guide</i> for details.
<code>printer</code>	Not supported by this device driver.
<code>startc c</code>	Sets the XON flow control character. The character may be given as a decimal, octal or hexadecimal number. Octal numbers are recognized by the presence of a leading zero, and hexadecimal numbers are denoted by a leading "0x". For example, the standard XON character, <CTRL-Q>, can be entered as "17" (decimal), "021" (octal) or "0x11" (hexadecimal).
<code>stopc c</code>	Sets the XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see <code>startc</code> , above, for format of octal and hexadecimal numbers).
<code>astartc c</code>	Sets auxiliary XON flow control character. The character may be given as a decimal, octal, or hexadecimal number (see <code>startc</code> , above, for format of octal and hexadecimal numbers).
<code>astopc c</code>	Sets auxiliary XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see <code>startc</code> , above, for format of octal and hexadecimal numbers).
<code>[-]aixon</code>	Enables auxiliary flow control, so that two unique characters are used for XON and XOFF. If both XOFF characters are received, transmission will not resume until both XON characters are received.
<code>maxcps n</code>	Sets the maximum Characters Per Second (CPS) rate at which characters are output to the transparent print device. The rate chosen should be just below the average print speed. If the number is too low, printer speed will be reduced. If the number is too high, the printer will resort to flow control, and user entry on the terminal will be correspondingly impaired. Default is 100 CPS.
<code>maxchar n</code>	Sets the maximum number of transparent print characters the driver will place in the output queue. Reducing this number increases system overhead; increasing this number delays operator keystroke echo times when the transparent printer is in use. Default is 50 characters.

<code>bufsize</code> <i>n</i>	Sets the driver's estimate of the size of the transparent printer's input buffer. After a period of inactivity, the driver bursts this many characters to the transparent printer before reducing to the <code>maxcps</code> rate selected above. Default is 100 characters.
<code>onstr</code> <i>s</i>	Sets the terminal escape sequence to turn transparent printing on. The string <i>s</i> can be composed of standard ASCII printing and non-printing characters; control (non-printing) characters must be entered by their octal values, and must consist of three digits preceded by a back-slash (“\”) character. For example, the “Escape” character <Esc>, 33 octal, should be entered as “\033”. Thus, if transparent printing is turned on by the string “<Esc>[5i” (ANSI standard), <i>s</i> should be entered as “\033[5i”.
<code>offstr</code> <i>s</i>	Sets the terminal escape sequence to turn transparent printing off. See “ <code>onstr</code> ”, above, for the format of the string <i>s</i> .
<code>term</code> <i>t</i>	Sets the transparent printer on/off strings to values found in the internal default table. Internal defaults are used for the following terminals: adm31 , ansi , dg200 , dg210 , hz1500 , mc5 , microterm , multiterm , pcterm , tvi , vp-a2 , vp-60 , vt52 , vt100 , vt220 , wyse30 , wyse50 , wyse60 , or wyse75 . If the terminal type is not found in the internal default table, then <code>ditty</code> reads the <code>terminfo</code> entry for the terminal type and sets transparent print on/off strings to values given by the mc5/mc4 attributes found there.

Also see `stty(1)`, `ioctl(2)`, `termio(4)`, and `terminfo(5)`.

Using the ditty Command with Parallel Ports

The ditty command features a special mode for use with the parallel port on Digi PORTS/8emp modules. When the `-p` flag is included in the ditty command, certain flags and functions are interpreted differently than for serial ports. Following is an example of the output of the command `ditty -p ttya25` (for System V, Release 4, the device name would be `/dev/term/a25`):

```
parallel output port
pulse width 0.8 us -noack hupcl
BUSY- IP0+ IP1- IP2- OP+
-opost -olcuc -onlcr -ocrnl -onocr -onlret
```

Pulse Width defines how long the data is strobed; the default is 0.8 microseconds, which is the fastest supported strobe. To change the strobe length, use the ditty command as follows:

```
ditty 1 ttya25
```

The above example sets a 1.0 microsecond pulse width.

Valid pulse widths are: 0.8, 1, 1.2, 1.6, 2, 3, 4, 5, 10, 20, 30, 40 and 50.

The “baud rate” of the port is used to control the pulse width—9600 baud or faster translates to 0.8 microseconds; 4800 baud is 1 microsecond, 2400 is 1.2 microseconds, etc. The `-p` flag is implied, and need not be included in the command.

`[-]noack`

When `noack` is specified (`ditty noack ttya25`), the device driver will not wait for an ACK pulse from the receiving device before transmitting another character; only the BUSY signal will stop data transmission.

When `noack` is turned off (`ditty -noack ttya25`), an ACK pulse is expected after each character is transmitted.

The `dsrpace` control is mapped to `noack` for parallel ports. The `-p` flag is implied, and need not be included in the command.

`hupcl` has the usual meaning: “hang up on close”. Since DTR is mapped to the parallel port’s SELECTIN line (OP; see below), dropping and raising DTR (which happens when a port is closed and then opened again) will have the effect of strobing the RESET line of the parallel device connected to the port. `-hupcl` prevents the device from being reset when it is reopened.

BUSY shows the state of the BUSY line. BUSY is not asserted (-) when the input device can accept data, and asserted (+) when it cannot. BUSY is mapped to DSR for parallel ports.

IP0, IP1 and IP2 are device dependent inputs. They provide information only, and do not affect the flow of data. They are typically used for signals like Paper Out, On/Off Line, etc. IP0, IP1 and IP2 are mapped to CTS, DCD and RI, respectively.

OP is the SELECTIN line, which is mapped from DTR. When OP changes from LOW to HIGH, the parallel port's RESET line is strobed. Normally, the RESET function is handled by the driver; however, a RESET can be forced by issuing the command "ditty OP- ttya25" followed by "ditty OP+ ttya09", or by the command "ditty reset ttya25". The -p flag is implied, and need not be included in the command.

`[-]opost, [-]olcuc, [-]onlcr, [-]ocrnl, [-]onocr, [-]onlret`

These flags have the usual meaning (see `stty(1)`).

DigiPRINT Transparent Print Feature

Description & Theory of Operation

Most terminals have an auxiliary port that can be connected to a serial printer. These terminals support two print modes, Auxiliary and Transparent. If both print modes are OFF, data received by the terminal is simply displayed on the screen. With Auxiliary print mode ON, data received by the terminal is displayed on the screen, and is also transmitted to the printer. With Transparent Print Mode ON, the terminal transmits data received directly to the printer, without displaying it on the screen.

DigiPRINT allows you to use your terminal in a normal manner, while information is also being sent *over the same serial connection from the host* to the printer connected to the terminal's auxiliary printer port. This is "transparent printing." The DigiPRINT software determines whether packets of data are bound for the screen or for the printer, and precedes data bound for the printer with the Transparent Print Mode ON command, and follows it with the Transparent Print Mode OFF command.

Data for the terminal screen has the highest priority, and DigiPRINT sends data to the printer only if there is a break in information being sent to the screen. If continuous data is being transmitted to the terminal device, nothing gets sent to the printer.

Whenever an auxiliary printer port is used, flow control to the printer becomes an issue. If the printer falls behind and invokes flow control, output to both the printer and the terminal is stopped: this is aggravating to the terminal user. The `ditty` command provides three parameters to limit printer output and avoid this situation. (See *Setting Terminal Options with ditty* on page 35 for a complete description of the `ditty` command.)

The parameter `maxcps` limits the maximum printer port character-per-second data rate. This number should be set to the minimum character rate the printer can sustain in typical use.

The parameter `maxchar` limits the number of characters queued to the printer ahead of terminal output. Lower numbers increase system overhead, higher numbers result in keystroke echo delays. A value of 50 is generally a good compromise at 9600 baud.

The parameter `bufsize` should be set to a value just below the printer's buffer size. After a period of inactivity, the driver will burst up to this many characters to the printer to fill the print buffer before slowing to the `maxcps` rate.

The printer on/off strings are also set using `ditty`.

DigiPRINT will be available for use after the Digi device driver software for your operating system is installed, and the transparent print options are activated with the `ditty` program.

A cable must be connected between the auxiliary port of the terminal and the printer. The baud rate on the terminal auxiliary port and the printer must be the same, and the printer and the auxiliary port of the terminal must use the same handshaking mode. The auxiliary port must also be enabled. If your terminal is not one of those directly supported, you must know the escape sequence of your terminal.

Refer to your terminal and printer manuals for connection information, escape codes, and to see what handshaking modes are supported (i.e. `xon/xoff`, `busy/ready`, `rts/cts`, etc.).

Printer devices (`pra01`, etc.) must not be in either the `/etc/inittab` or `/etc/ttys` files, and must not be enabled.

Transparent Print Activation

DigiPRINT is activated with `ditty`. The `ditty` program configures the Digi tty device driver for transparent print options. (See *Setting Terminal Options with ditty* for a complete description of the `ditty` command.) The `ditty` command must be run each time the machine is booted. Usually, the best way to do this is by adding `ditty` commands to your `etc/rc` system initialization file. (Put them in a text file in the directory `/etc/rc.d/8`—See your System Administrators Guide for details.) Alternatively, you may include the `ditty` command sequence in your `.login` or `.profile` files, to ensure that DigiPRINT is activated when you log in. Your System Administrator can help you edit these files. The pathname for `ditty` is `/usr/bin/ditty`

DigiPRINT transparent print Options are set using the `ditty` program in the following manner:

```
ditty[ -a ] [ option(s) ] ttyname
```

The command line options are:

<code>maxcps n</code>	Sets the maximum Characters Per Second (CPS) rate at which characters are output to the transparent print device. See <i>Setting Terminal Options with ditty</i> (page 35) for more information.
<code>maxchar n</code>	Sets the maximum number of transparent print characters the driver will place in the output queue. See <i>Setting Terminal Options with ditty</i> for more information.
<code>bufsize n</code>	Sets the driver's estimate of the size of the transparent printer's input buffer. See <i>Setting Terminal Options with ditty</i> for more information.
<code>onstr s</code>	Sets the terminal escape sequence to turn the transparent printer on. An arbitrary octal character <code>xxx</code> may be given as <code>\xxx</code> .
<code>offstr s</code>	Sets the terminal escape sequence to turn the transparent printer off. An arbitrary octal character <code>xxx</code> may be given as <code>\xxx</code> .
<code>term t</code>	Specifies the terminal type. See <i>Setting Terminal Options with ditty</i> for more information.
<code>ttyname</code>	Specifies the tty device.

Now data can be sent to your printer via the printer devices.

Example: `cat filename > /dev/prn01`

ditty Examples

Example 1:

The following command configures the DigiPRINT options for a DEC VT100 terminal connected to `/dev/ttya01`. (Note that the printer uses `/dev/prn01`). `maxcps`, `maxchar` and `bufsize` are left to defaults. Type:

```
ditty term vt100 /dev/ttya01
```

Example 2:

The following example uses `onstr` and `offstr` arguments. (This sets the terminal to use ANSI Standard.) Again `maxcps`, `maxchar`, and `bufsize` are defaults. Type:

```
ditty onstr "\033[5i" offstr "\033[4i" /dev/ttya01
```

Example 3:

This example command sets the DigiPRINT option for a WYSE30 terminal, with `maxcps` of 75, a `maxchar` of 100, and a printer buffer size, `bufsize`, of 1000. Type (all on one line, with a carriage return at the end only):

```
ditty term wyse30 maxcps 75 maxchar 100 bufsize 1000 /dev/ttya01
```

DigiSCREEN Multiple Screen Utility

Description and Theory of Operation

DigiSCREEN is a utility for SCO UNIX systems that allows a single physical terminal to be connected to several virtual terminal sessions (*screens*) at one time. It is mainly intended for use with terminals that have two or more pages of screen memory. With such terminals, switching between virtual screens will also switch between physical terminal screen pages, allowing each virtual screen's image to be saved and restored. On terminals without multiple pages of screen memory, DigiSCREEN can still be used to switch among virtual screen sessions, although the appearance of the screen will not be maintained when switching screens.

Note: For full support of DigiSCREEN, your terminal must be able to switch internal screen pages on command *and must remember the cursor position for each page*. While DigiSCREEN will work on both smart and dumb terminals, screen images are not saved during screen changes on dumb terminals. DigiSCREEN also supports terminals connected to two or more computers through separate serial ports.



DigiSCREEN is supported only by the SCO UNIX device driver. It is *not* supported by the System V, Release 4 device driver.

Options

DigiSCREEN is called with the following format:

```
dscreen[-i infile] [-t termtype]
```

If *infile* is specified, it will be used as the source of terminal configuration information. Otherwise, if the environment variable DSINFO is defined, it specifies the name of the file to be used as the source of terminal configuration information. If neither *infile* nor DSINFO is specified, the configuration information is read from the file `/etc/dsinfo`. This option is used to define a different set of keys to be used with DigiSCREEN, e.g. when the originally defined DigiSCREEN keys conflict with an application one wishes to use.

The terminal type is used to select which entry in the *infile* (default `/etc/dsinfo`) is used to describe the terminal. If the desired terminal type does not match the setting of the TERM environment variable (again, for alternate key mappings), it can be specified as *termtype* with the `-t` option.

Using DigiSCREEN

When DigiSCREEN is run, it starts up one virtual screen. Some of the keys on the terminal keyboard will not be passed through to the virtual screen; instead, DigiSCREEN will intercept these keys and perform certain actions when they are pressed. The actions include select a specific screen, block all input and output, start a new screen, end DigiSCREEN (exit code 0), quit DigiSCREEN (exit code 1), switch to the previous screen, and list the DigiSCREEN keys and what they do. Which function each key performs is dependent upon the terminal and the terminal description in the `dsinfo` file.

When a new virtual screen is created, it is assigned to a select key. When this key is pressed, DigiSCREEN will switch the physical terminal to the video page associated with the particular virtual screen and direct all input and output to go between the physical terminal and the virtual screen. Each virtual screen must have a select key; once all of the select keys defined in the `dsinfo` file have virtual screens assigned to them, no more screens may be created. Individual screen sessions will end when the original shell process exits, and this will free the associated select key for use with another virtual screen. DigiSCREEN exits when there are no more active screens.

Block keys can be used to stop output (in a fashion similar to `<Ctrl-S>` when using “ixon” flow control). However, the true purpose of these keys is to allow for transparently setting up terminal sessions on two computers using a terminal that has two serial ports. See `dsinfo` (page 49) for more information.

Pressing a new screen key will create a new screen and assign it to one of the select keys, unless one of the necessary resources is exhausted. Each new screen requires a select key as defined in the `dsinfo` file, a DigiSCREEN pseudo terminal device, enough memory for the various structures used to keep track of the screen, and a process to run the shell. If any of these are not available, the new screen operation will fail and print a message indicating the reason for the failure.

Pressing an end key will send a `SIGHUP` signal to all the screen sessions, clean up, and exit with a status of 0. Pressing a quit key will perform the same actions, but will exit with a status of 1.

Pressing a previous key will switch the terminal to the screen that was last displayed.

Pressing a list key will cause a list of the keys recognized by DigiSCREEN and their actions to be displayed on the terminal. When DigiSCREEN starts a new screen, it will display the message “Press *KEY* for help” (where *KEY* is the name of the list key) if there is a list key defined.

Dynamic Screen Assignment

Normally, the terminal description entry in the `dsinfo` file will have the same number of screen selection keys as the terminal has physical screen pages. However, if more screen selection keys are defined than the number of physical screen pages defined, DigiSCREEN will dynamically assign physical screen pages to virtual screens. When a virtual screen that doesn't have an associated page of screen memory in the terminal is selected, DigiSCREEN assigns the least recently used physical screen to the virtual screen. When this occurs, some sort of indication is given that the physical screen is connected to a different virtual screen; for instance, the screen may be cleared. Using a terminal that has only one physical screen is the simplest case of this; the one screen is shared between all virtual screens.

Notes:

- Avoid switching screens when the screen is being written to; you may interrupt an escape sequence and leave the terminal in an unknown state.
- Even if your terminal saves the cursor position for individual screens, it may not save other states such as insert mode, inverse video, etc. If this is the case in your situation, make sure you are not in any such mode when you switch screens.

DSINFO DigiSCREEN Information File

Description

`/etc/dsinfo` is a database of terminal descriptions used by `dscreen` (DigiSCREEN). The information in the descriptions include what keys are to be used by DigiSCREEN and what functions they perform, how many pages of screen memory the terminal has, and what code sequences are sent/received to use these features.

Entry Format

Entries in `dsinfo` consist of a number of comma separated fields. The first field is a list of alternate names for the terminal, separated by “|” characters.

The remaining fields are strings describing the capabilities of the terminal to DigiSCREEN. Within these strings, the following escape codes are recognized:

<code>\E</code> , <code>\e</code>	escape character
<code>\n</code> , <code>\l</code>	newline (a.k.a. linefeed) character
<code>\r</code>	carriage return
<code>\t</code>	tab character
<code>\b</code>	backspace character
<code>\f</code>	formfeed character
<code>\s</code>	space character
<code>\nnn</code>	character with octal value <i>nnn</i>
<code>^x</code>	<Ctrl- <i>x</i> > for any appropriate <i>x</i>

Any other character preceded by a backslash will yield the character itself. The strings are entered as `type=string`, where `type` is the type of string as listed below, and `string` is the string value.

String Types

The string types are as follows:

`dskx` A string type that starts with “`dsk`” describes a key. The type must be four letters long, and the fourth letter *x* indicates what action is taken when the key is received. The key types are:

Type	Action
<code>dsk s</code>	Switch Screens
<code>dsk b</code>	Block Input and Output
<code>dsk e</code>	End DigiSCREEN
<code>dsk q</code>	Quit DigiSCREEN (non-zero exit status)
<code>dsk c</code>	Create New Screen
<code>dsk p</code>	Switch to Previous Screen
<code>dsk l</code>	List Keys and Actions

Any other key type (a string type `dskx` that doesn't end in *s*, *b*, *e*, *q*, *p*, or *l*) will cause no internal `dscreen` action, but will show up in the key listing and will be recognized and acted upon (see below). A type of `dskn` (*n* for No Operation) is guaranteed not to be used for any function in future versions; it is recommended that this be used when no internal `dscreen` action is desired. The value string for each key has three substrings, which are separated by “|” characters (use “\|” to include the “|” character in one of the substrings). The first substring is the sequence of characters that the terminal sends when the key is pressed. The second substring is a label for the key that is printed when a list of the keys is presented (for example, “Shift-F1”). The third substring is a sequence of characters that DigiSCREEN sends to the terminal when this key is pressed, before performing the action this key requests.

`dsp` A string type of “`dsp`” describes a physical screen in the terminal. One `dsp` string should be present for each physical screen in the terminal. The value string for each physical screen has two substrings, which are separated by a “|” character (again, use “\|” to include the “|” character in one of the substrings). The first substring should be the sequence of characters to send to the terminal to display and output to the particular physical page on the terminal. The second substring is sent to the terminal any time the page is used for something new. This second substring is usually set to the clear screen sequence. It is sent under two conditions. The first condition is when a new virtual terminal session

is being created. The second condition occurs when the user is running more virtual terminals than there are physical screens; if the user selects a virtual terminal such that DigiSCREEN has to re-use one of the physical screens, it will send this sequence to the screen to indicate to the user that the screen contents don't match the output of the virtual terminal to which it is connected. Note that running with more virtual terminals than physical screens can be quite confusing and is not particularly recommended; it can be avoided by defining no more screen selection keys ("dsk=...") than physical screens ("dsp=...") in the dsinfo entry.

dst A string with a type of "dst" adjusts DigiSCREEN's input timeout. The value of the string should be a decimal number. The timeout value is in tenths of a second, and has a maximum value of 255. The default timeout value is 1 (or .1 seconds). When DigiSCREEN recognizes a prefix of an input key sequence but doesn't have all the characters of the sequence, it waits for more characters. If the timeout occurs before more characters are received, the characters are sent on to the virtual screen and DigiSCREEN will not consider these characters as part of an input key sequence. It may be necessary to raise this value if one or more of the "keys" DigiSCREEN is to trigger on is actually a number of keystrokes (i.e. assigning Ctrl-Z 1, Ctrl-Z 2, Ctrl-Z 3, etc. for screen selection, Ctrl-Z N for new screen and so on).

Example 1

The following example entry is for a Wyse 60 with three screens:

```
wy60|wyse60|wyse model 60,  
      dsks=^A^M|Shift-F1|,  
      dsks=^Aa^M|Shift-F2|,  
      dsks=^Ab^M|Shift-F3|,  
      dskc=\200|Ctrl-F1|,  
      dske=\201|Ctrl-F2|\Ew0\E+ ,  
      dskl=\202|Ctrl-F3| ,  
      dsp=\Ew0|\E+ ,  
      dsp=\Ew1|\E+ ,  
      dsp=\Ew2|\E+ ,
```

With this entry, <Shift-F1> through <Shift-F3> are used for selecting screens 1 through 3, respectively. <Ctrl-F1> will create a new screen, <Ctrl-F2> will send “<ESC> w 0 <ESC> +” to the screen (switching to window 0 and clearing the screen) and then end dscreen, and <Ctrl-F3> will list the keys and their functions. The three physical screens are displayed by sending “<ESC> w 0”, “<ESC> w 1”, and “<ESC> w 2.” Each time a physical screen is used for a new screen, the sequence “<ESC> +” will be sent to the terminal, which will clear the screen.

Example 2

This example is, again, for a Wyse 60 with three screens, but one of the screens is on a second computer communicating through the second serial port on the terminal:

```
wy60-1|wyse60-1|wyse model 60 - first serial port,
      dskb=^A^M|Shift -F1|,
      dskb=^Aa^M|Shift -F2|,
      dskb=^Ab^M|Shift -F3|\Ed#^Ab\r^T\Ee9,
      dskc=\200|Ctrl -F1|,
      dske=\201|Ctrl -F2|\Ed#\201^T\Ew0\E+,
      dskl=\202|Ctrl -F3|,
      dsp=\Ew0|\E+,dsp=\Ew1|\E+,

wy60-2|wyse60-2|wyse model 60 - second serial port,
      dskb=^A^M|Shift -F1|\Ed#^A\r^T\Ee8,
      dskb=^Aa^M|Shift -F2|\Ed#^Aa\r^T\Ee8,
      dskb=^Ab^M|Shift -F3|,
      dskc=\200|Ctrl -F1|,
      dske=\201|Ctrl -F2|\Ed#\201^T\Ew0\E+,
      dskl=\202|Ctrl -F3|,
      dsp=\Ew2|\E+,
```

For this setup to work, DigiSCREEN must be run on both computers, with terminal type `wy60-1` on the first computer and terminal type `wy60-2` on the second computer (using the `-t` option to DigiSCREEN). The `wy60-1` entry will be examined first.

The first two key entries are unchanged from the original `wy60` entry. The third key, however, has type “`dskb`,” which means block both input and output. When this key is pressed, the sequence “`<ESC> d # <Ctrl-A> b <CR> <Ctrl-T> <ESC> e 9`” is sent to the terminal; after this output is blocked and DigiSCREEN continues scanning input for key sequences but discards all other input.

The effects caused by the sequence sent to the terminal contain the real magic here. The sequence “`<ESC> d #`” puts the terminal in “Transparent Print Mode,” which echoes all characters up to a `<Ctrl-T>` out the other serial port. The characters “`<Ctrl-A> b <CR>`” are sent out the other serial port, informing the `dscreen` process on the other computer that it should activate the window associated with the `<Shift-F3>` key. The “`<Ctrl-T>`” takes the terminal out of the Transparent Print mode, and the sequence “`<ESC> e 9`” tells the terminal to switch to the other (“AUX”) serial port for data communications.

At this point the other computer takes over and sends an “<ESC> w 2” to switch to the third physical screen, and then resumes normal communication.

The wy60-2 entry follows the same general pattern for keys <Shift-F1> and <Shift-F2>: switch to transparent print mode; send function key string to other computer; switch transparent print off; and switch to the other serial port. The end key (<Ctrl-F2>) works the same for both computers; it sends the end key sequence to the other computer through the transparent print mechanism, switches the terminal to window 0, clears the screen, then exits.

Error Messages

The following error messages are generated by the driver:

```
***WARNING*** Undefined message type (12, 13, 14).
***WARNING*** Undefined misc. command.
***WARNING*** Undefined command.
***WARNING*** Unrecognized ID response.
***WARNING*** Sent more data than authorized.
***WARNING*** Access to unopened port.
***WARNING*** Window overrun.
***WARNING*** Unexpected open response.
***WARNING*** Peer reset connect.
***WARNING*** Close of unopened port.
***WARNING*** Unimplemented open.
```

If you get any of these messages, the connection has been reset; the daemon must be restarted.

To restart the daemon, log in as root and enter the following command:

```
sh /etc/idrc.d/ncxa
```



If more than one daemon is running, there is no need to kill the running daemons. They will not be affected by the above command.

Digi Support Services

The Digi Bulletin Board System

Digi provides an electronic bulletin board service (BBS) for our customers. This bulletin board provides general and technical information about Digi's products.

The Digi BBS allows users to download software drivers as soon as they become available. There is also a feature to allow users with problems or questions about Digi products to leave messages to Digi Technical Support.

Using the Digi BBS is easy. Simply dial **(612) 943-0550**. In Europe, dial **+49 221 9205211**; in Asia, dial **+65 735 2460**. The bulletin board accepts calls at 1200, 2400, 9600 and 14.4K baud. V.32, HST 14.4, V.42 and V.42*bis* standards are supported, with full MNP class 1-5 error correction and data compression.

The recommended modem communications parameters are 8 bits, no parity and one stop bit (**8 N 1**). Other settings may also work.

Download protocols include Zmodem, Xmodem, Ymodem, Kermit and others.

Internet FTP Server

Digi has set up an Anonymous FTP server for those with access to the Internet network. The address is `ftp.digibd.com`. Log in as `anonymous`, and enter your E-mail address when asked for a password. Drivers and installation tips are located in the `/drivers` directory. A text file, `download.doc`, gives information on uncompressing the files after downloading. Tip: Be sure to enter "bin" before downloading, to ensure binary transfer of files.

World Wide Web Server

Product information, manuals, new product announcements, programs, application stories and more can be obtained through the World Wide Web. Our address is `http://www.digibd.com`

DigiFACTs FaxBack Server

Manuals and technical information can also be obtained by FAX. To use the FaxBack server, simply call (612) 943-0573 on a touch tone phone.

Information About Your System

Serial number of your Digi product: _____

Make, model and clock speed of your computer: _____

How much RAM does your computer have? _____

**Hard disk
controller:**

Type: _____ Memory addressed at: _____

I/O port used: _____ IRQ: _____

LAN card: Type: _____ Memory addressed at: _____

I/O port used: _____ IRQ: _____

Other: Type: _____ Memory addressed at: _____

I/O port used: _____ IRQ: _____

Operating system: _____ Version: _____

Digi device driver version: _____

Technical Support

At Digi, we are proud of our products, and support them. Our dealers and distributors are fully trained on our product line, so that they can help you on a technical level should assistance be needed.

Your first level of support is your Digi dealer, the place where you purchased your Digi product. Your dealer has the training to help you with any installation questions or difficulties you might have.

If you still experience difficulties (after contacting your first level of support), Digi has a staff of Technical Support Specialists that can assist you. They can be reached at **(612) 943-0578**. In Europe, call **+49 221 920520**, and in Asia, call **+65 732 1318**. FAX numbers are: (612) 943-0579 (USA), +49 221 9205210 (Europe) and +65 732 1312 (Asia).

When you call Digi Technical Support, please call from a position where you can operate your system. Also, please fill out the form on the facing page before calling, so your Technical Support representative can have a clear picture of your system and any potential conflicts between devices.

Digi Technical Support can also be reached via Internet E-mail. Please send correspondences to support@digibd.com, and include your voice and FAX phone numbers.

Customer Service

Digi also has a staff of Customer Service representatives to help you with software and documentation update requests, as well as Returned Merchandise Authorizations (RMAs) in case you need to return your Digi product for repair (see page 60). They can be reached at **(612) 943-0577**.

Digi Customer Service can also be reached via Internet E-mail. Please send correspondences to cust_serv@digibd.com, and include your voice and FAX phone numbers.

Return Procedures

All Digi products have a five-year parts and labor warranty, and we are ultimately responsible for any defective parts, according to the limits specified in the warranty. However, many of the reported problems are due to factors other than defects in the product itself. To save you time and possibly additional cost, Digi asks that you *first* try to resolve any difficulties by contacting our Technical Support representatives at **(612) 943-0578**.



Be sure to have the serial number of your Digi product at hand before calling Technical Support.

Returns should be directed to the dealer or distributor from whom you purchased the product. If you need to return your Digi product for repair, it is first necessary to obtain an RMA (Returned Merchandise Authorization) number from Digi, by speaking to a Digi Customer Service representative. Authorized returns should be shipped to Digi International, 10000 West 76th Street, Eden Prairie, MN 55344. The RMA number should appear on the shipping carton, on or near the address label.



Products received without an RMA number clearly marked on the outside of the package will be returned, unopened, to the sender

