Software Manual

MenuPort Interface™ (MPI)

SCO UNIX
UNIX System V Release 4

92000154 Rev K

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

The Digi device driver software for UNIX is installed by the Digi MenuPort Interface™ (MPI™) package; the software development system is *not* required to create a new kernel. The link kit, however, must be installed.

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 27)
- DigiPRINT™ transparent printing (page 34)

In addition to the actual device drivers, the MenuPort Interface package includes Digi Port Authority™ (also called DPA™) a tool for monitoring and testing Digi intelligent products, and DigiSCREEN™, a utility that allows a single physical terminal to be connected to several virtual terminal sessions (*screens*) at one time. Instructions for using Digi Port Authority are provided in help screens; for information on using DigiSCREEN, refer to the man pages **`DSCREEN(1)`** and **`DSINFO(4)`** (enter **`man dscreen`** or **`man dsinfo`** at a UNIX prompt).

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

Supported Products

The MenuPort Interface package supports the Digi AccelePort Xe, Xr, C/X, EPC/X and Xem families, and Avanstar PCI adapters under System V Release 4 (including SCO UnixWare 1.1 and 2.x, and NCR MPRAS 3.0.x), SCO UNIX 3.2.x and SCO OpenServer 5.0.

Software Installation

Installation of the Digi device drivers is a two-part procedure:

First, the Digi MenuPort Interface (MPI) package is installed onto your hard disk. Second, the MPI package is run and the appropriate device drivers for your system are installed and configured.

Note: 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 MenuPort Interface (MPI) Package

SCO UNIX

1. Log in as super-user (root).
2. Run **`custom`**. Select “New Product”. Insert the Digi MenuPort Interface device driver diskette when asked to do so.
3. When the MPI installation is complete, quit **`custom`**.
4. Now skip to page 7 to run the MPI program.

System V, Release 4

1. Log onto the console as super-user (root).
2. To begin the installation, insert the Digi MenuPort Interface device driver diskette into drive A and enter:

   ```
   pkgadd -d diskette1
   ```

   (If you wish to load the driver package from drive B, substitute “**`diskette2`**” for “**`diskette1`**” in the above command.)

   Alternatively, you can use the **`pkgask`** program to create a response file (useful for configuring multiple systems). Enter:

   ```
   pkgask -d diskette1 -r filename
   ```

   where *filename* is the name of the response file. Follow the online help screens for more information on configuring adapters. If you already have a response file from a previous **`pkgask`** session, you can use it by entering:

   ```
   pkgadd -d diskette1 -r filename
   ```

   MPI will be installed along with the drivers. MPI will not have to be run to configure the adapters.
3. Now proceed to page 7 to run the MPI program.

Running the MenuPort Interface (MPI) Program

To run the MenuPort Interface program, log in as root and enter:

```
mpi
```

After the restricted rights legend, you will see the Main menu. Items in this menu are:

- **Config**, which lets you install, change and remove device drivers for specific adapters and operating systems.
- **System**, which lets you build a new kernel (SCO only), load and unload drivers (SVR4 only), update installed device drivers and create an image of the kernel.
- **Monitor**, which runs Digi Port Authority (DPA), a software diagnostic tool that allows users to easily monitor the status of the Front End Processor/Operating System (FEP/OS), which runs on all Digi intelligent serial port adapters. The DPA software also shows the status of the individual ports. With DPA, system managers can tell instantly if an adapter is no longer working, if its performance level decreases, of if a port is non-functional, enabling them to work more productively and solve system errors more quickly. DPA also provides a loop back test to confirm that the adapter can transmit and receive data.
- **Port Configuration**, which provides an easy way to set up all Digi ports in your system without having to be an expert in UNIX system administration commands.
- **Notes**, which displays the release notes for the supported device drivers.

Note: MPI features context sensitive help screens; press <F1> at any time for an explanation of the currently displayed information and choices.

Device Names

SCO UNIX

AccelePort Xe Family Adapters

Format used for port names:

`/dev/ttyi`*bp*, **`/dev/pri`***bp*

`ttyi` Base name.

`pri` Base name for DigiPRINT transparent printing (see page 34).

b Adapter number (**`1-4`**).

p Port identifier:

`a-p` (lower case) are used for non modem control ports. These ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

`A-P` (upper case) are used for modem control ports. These ports require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with modems.

For example, Port 4 of Adapter 2 will be **`/dev/ttyi2d`**. The same port with modem control enabled would be named **`/dev/ttyi2D`**. The corresponding DigiPRINT devices are **`/dev/pri2d`** and **`/dev/pri2D`**.

SCO UNIX (continued)

AccelePort Xr Family Adapters

Format used for port names:

`/dev/ttyr`*bp*, **`/dev/prr`***bp*

`ttyr` Base name.

`prr` Base name for DigiPRINT transparent printing (see page 34).

b Adapter number (**`1-4`**).

p Port identifier:

`a-p` (lower case) are used for non modem control ports. These ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

`A-P` (upper case) are used for modem control ports. These ports require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with modems.

For example, Port 4 of Adapter 2 will be **`/dev/ttyr2d`**. The same port with modem control enabled would be named **`/dev/ttyr2D`**. The corresponding DigiPRINT devices are **`/dev/prr2d`** and **`/dev/prr2D`**.

SCO UNIX (continued)

AccelePort C/X, EPC/X and Xem Systems

Format used for port names:

`/dev/tty`*cp*, `/dev/pr`*cp*

`tty` Base name.

`pr` Base name for DigiPRINT transparent printing (see page 34).

c Concentrator or EBI module letter ID:

`a-z` (lower case) are used for non modem control ports. These ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

`A-Z` (upper case) are used for modem control ports. These ports require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with modems.

p Port number on the individual concentrator or EBI module (`01-64)`. Systems with a large number of ports may have to use some concentrator/EBI ID numbers more than once. When this occurs, the first module to use an ID will be assigned port numbers 01-16; the second module will be assigned 17-32, etc.

For example, Port 4 on concentrator or PORTS module “`b`” will be `/dev/ttyb04`. The same port with modem control enabled would be named `/dev/ttyB04`. The corresponding DigiPRINT devices are `/dev/prb04` and `/dev/prB04`.

SCO UNIX (continued)

Avanstar PCI Adapter

Format used for port names:

`/dev/tty`*tbp*, **`/dev/ttp`***tbp*

`tty` Base name.

`ttp` Base name for DigiPRINT transparent printing (see page 34).

t Type of port (**`a`** or **`A`**). Indicates whether port is a “standard” (**`a`**) or “modem control” (**`A`**) port.

Standard (non modem control) ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

Modem ports require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with modems.

b Adapter number:

`00` = adapter 1
`01` = adapter 2
`02` = adapter 3
`03` = adapter 4

p Port identifier:

`0` = port 1
`1` = port 2
`2` = port 3
`3` = port 4
`4` = port 5
`5` = port 6
`6` = port 7
`7` = port 8
`8` = port 9
`9` = port 10
`a` = port 11
`b` = port 12
`c` = port 13
`d` = port 14
`e` = port 15
`f` = port 16

NOTE: Ports 9 through 16 apply to 16-port adapters only.

For example, standard tty Port 4 of Adapter 2 would be **`/dev/ttya13`**. The same port with modem control enabled would be named **`/dev/ttyA13`**. The corresponding DigiPRINT devices would be **`/dev/ttypa13`** and **`/dev/ttypA13`**.

System V Release 4

AccelePort Xe Family Adapters

Format used for port names:

`/dev/term/i`*bp*, **`/dev/pri`***bp*

`i` Base name.

`pri` Base name for DigiPRINT transparent printing (see page 34).

b Adapter number (**`1-4`**).

p Port identifier:

`a-p` (lower case) are used for non modem control ports. These ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

`A-P` (upper case) are used for modem control ports. These ports require DCD (Data Carrier Detect) to be asserted before they can transmit data, and software flow control (see **`stty rtsxoff`** and **`stty ctsxon`**, in your System V Release 4 documentation) is enabled by default. Primarily used with modems.

For example, Port 4 of Adapter 2 will be **`/dev/term/i2d`**. The same port with modem control enabled would be named **`/dev/term/i2D`**. The corresponding DigiPRINT devices are **`/dev/pri2d`** and **`/dev/pri2D`**.

System V Release 4 (continued)

AccelePort Xr Family Adapters

Format used for port names:

`/dev/term/r`*bp*, `/dev/prr`*bp*

`r` Base name.

`prr` Base name for DigiPRINT transparent printing (see page 34). Transparent printer devices are placed in `/dev`.

b Adapter number (`1-4`).

p Port identifier:

`a-p` (lower case) are used for non modem control ports. These ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

`A-P` (upper case) are used for modem control ports. These ports require DCD (Data Carrier Detect) to be asserted before they can transmit data, and software flow control (see `stty rtsxoff` and `stty ctsxon`, in your System V Release 4 documentation) is enabled by default. Primarily used with modems.

For example, Port 4 of Adapter 2 will be `/dev/term/r2d`. The same port with modem control enabled would be named `/dev/term/r2D`. The corresponding DigiPRINT devices are `/dev/prr2d` and `/dev/prr2D`.

System V Release 4 (continued)

AccelePort C/X, EPC/X and Xem Systems

Format used for port names:

/dev/term/*cpt*, **/dev/pr***cpt*

/dev/term/
Directory path for tty devices.

pr Base name for DigiPRINT transparent printing (see page 34). Transparent printer devices are placed in **/dev**.

c Concentrator or EBI module letter ID Legal letters are **a-z**.

p Port number on the individual concentrator or EBI module (**01-64**). Systems with a large number of ports may have to use some concentrator/EBI ID numbers more than once. When this occurs, the first module to use an ID will be assigned port numbers 01-16; the second module will be assigned 17-32, etc.

t Type of port (**s** or **m**). Indicates whether port is a “standard” (**s**) or “modem control” (**m**) port.

Standard (non modem control) ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

Modem ports require DCD (Data Carrier Detect) to be asserted before they can transmit data, and software flow control (see **stty rtsxoff** and **stty ctsxon**, in your System V Release 4 documentation) is enabled by default. Primarily used with modems.

For example, standard tty port 4 on concentrator or EBI module “**b**” will be **/dev/term/b04s**. The same port with modem control enabled would be named **/dev/term/b04m**. The corresponding DigiPRINT devices are **/dev/prb04s** and **/dev/prb04m**.

System V Release 4 (continued)

Avanstar PCI Adapter

Format used for port names:

`/dev/term/a`*bpt*, `/dev/pra`*bpt*

`a` Base name.

`pra` Base name for DigiPRINT transparent printing (see page 34). Transparent printer devices are placed in `/dev`.

b Adapter number:

`00` = adapter 1
`01` = adapter 2
`02` = adapter 3
`03` = adapter 4

p Port identifier:

`0` =port 1	`4` = port 5	`8` = port 9	`c` = port 13
`1` = port 2	`5` = port 6	`9` = port 10	`d` = port 14
`2` = port 3	`6` = port 7	`a` = port 11	`e` = port 15
`3` = port 4	`7` = port 8	`b` = port 12	`f` = port 16

NOTE: Ports 9 through 16 apply to 16-port adapters only.

t Type of port (`s` or `m`). Indicates whether port is a “standard” (`s`) or “modem control” (`m`) port.

Standard (non modem control) ports do *not* require DCD (Data Carrier Detect) to be asserted before they can transmit data. Primarily used with directly connected terminals and printers.

Modem ports require DCD (Data Carrier Detect) to be asserted before they can transmit data, and software flow control (see `stty rtsxoff` and `stty ctsxon`, in your System V Release 4 documentation) is enabled by default. Primarily used with modems.

For example, standard tty Port 4 on Adapter 2 is `/dev/term/a13s`. The same port with modem control enabled would be named `/dev/term/a13m`. The corresponding DigiPRINT devices are `/dev/pra13s` and `/dev/pra13m`.

Enabling the New Ports

SCO UNIX

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/ttyi1a
```

(Assuming the terminal is connected to **`ttyi1a`**.)

(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 everything looks good, but you are still not getting output, run the loop back tests described on page 21.

  If a port fails, check another port. If more than one port fails, the problem is probably due to memory contention, rather than defective hardware. Try reinstalling the device driver with a different memory start address.
- 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).

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

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 port ID is a capital letter; e.g. **`ttyi1A`**) 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 must be defeated, either by “tying it high” (e.g. connecting it to the same port’s DTR line) or by issuing the command **`ditty forcedcd ttyi1A`** (assuming that the port in question is **`ttyi1A`**).

Another way to bypass the DCD requirement is to use non modem control devices (where the port ID is a small letter; e.g. **`ttyi1a`**). Non modem control devices function identically to modem control devices, except that they do not require DCD to be asserted 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 ttyi1a
```

The above command will cause a login prompt to be sent to the terminal connected to **`ttyi1a`**. To activate the other ports, repeat the enable command with the appropriate port name.

Note:

As an alternative to step 3, you can use the Port Configuration Utility of MPI to enable ports. See *Running MPI*, on page 7.

System V Release 4

Please refer to your UNIX System Administrator's Guide, under the *Peripherals Setup* section, for the details on how to enable serial ports. The following example shows how to set up serial ports for use with terminals:

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/i1a
```

(Assuming the terminal is connected to **i1a**.)

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 terminals is 9600 baud, 7 data bits, even 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 everything looks good, but you are still not getting output, run the loop back tests described on page 21.

  If a port fails, check another port. If more than one port fails, the problem is probably due to memory contention, rather than defective hardware. Try reinstalling the device driver with a different memory start address.
- 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, 7 data bits, 1 stop bit, and even parity).

Once you can redirect output to your screen with the test above, perform the following steps to enable the ports for monitoring by **ttymon**.

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).

4. 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/i1a`** and **`/dev/term/i1A`**).

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

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

Note: If you performed the above steps, but the ports are not enabled, you can enable the ports from the command line. The following example is for **`/dev/term/i1a`**:

```
pmadm -a -p ttymon3 -s i1a -i root -f u -v `ttyadm -V` \
-m "`ttyadm -d /dev/term/i1a -l 9600 -s /usr/bin/login`"
```

where **`ttymon3`** is the tag for the port monitor and **`i1a`** is the service tag. The **`-m`** argument is the monitor-specific information, **`/dev/term/i1a`** is the device with 9600 as the label in **`/etc/ttydefs`**. Substitute each device name for **`i1a`** as appropriate.

Note: The default UNIX SYSTEM V REL 4 settings for these ports are: **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**.

Note: As an alternative to steps 3-6, you can use the Port Configuration Utility of MPI to enable ports. See *Running MPI*, on page 7 (available in MPI version 2.3.0 or greater).

Serial Printers

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

`ditty printer` [*baud rate and flow control options*] **`ttyi1a`**

This command forces the specified port to stay open, thus preserving any "non-sticky" serial line characteristics (see *Setting Terminal Options with ditty*, beginning on page 27, 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 a text file in the directory **`/etc/rc.d/8`** if you are running SCO UNIX—or in **`/etc/rc2.d`** if you are running System V Release 4—so that it is automatically executed whenever the system is booted.

EXAMPLE:

The command shown below results in 9600,8,1,N, with software flow control.

```
ditty printer ixon opost onlcr ttyi1a
```

Note: As an alternative to the above procedure, you can use the Port Configuration Utility of MPI to set up serial printers. See *Running MPI*, on page 7 (available in MPI version 2.3.0 or greater).

Loop Back Test

Individual ports can be tested for correct operation by running the Digi Port Authority (DPA) program. To perform port tests with the DPA program, perform the following steps:

1. Run the MenuPort Interface program by entering the following command:

```
mpi
```

2. From the MenuPort Interface program's Main menu, select **Monitor**. The DPA program will be run.

3. From the DPA program's Main menu, use the up and down arrow keys select the adapter you wish to test.

NOTE: If the arrow keys do not map correctly, the DPA program will display the following message:

```
Invalid Key. (Key Number)
```

If this occurs, substitute the standard UNIX cursor movement keys for the up and down arrows: Down = **`J`** or **`j`**; Up = **`K`** or **`k`**.

4. If the adapter you selected was an Xe, Xr or Avanstar adapter, you will now see the Channel Monitoring screen. Proceed to Step 5.

If the adapter you selected was a C/X, EPC/X or Xem host adapter, you will now see the Adapter Configuration screen. Use the arrow keys to select the concentrator or EBI module you wish to test, and press <Enter> to get to the Channel Monitoring screen.

5. From the Channel Monitoring screen, use the left and right arrow keys to select the channel to test (or press **`1-9`** for channels 1-9, respectively, and **`a-g`** for channels 10-16, respectively).

A loop back diagnostic test can now be run against the selected channel by pressing the **T** (or **t**) key from the channel monitoring screen. A loop back plug (see page 24 for specifications and part numbers) must be installed in the channel to be tested.

The loop back test consists of five phases:

Phase 1 Verify that the channel is not currently in use. All signals must be low.

If the port is busy, a message similar to the following will appear:

```
**** Port is Busy :DTR:CD:DSR:RTS
```

Probable cause of failure:
- Port is open by some process (e.g. getty).
- Incorrect loop back connector or cable.

Phase 2 128 bytes of test data (the letter **A** is used) are put into the transmit buffer.

If the write operation fails, a message similar to the following will appear:

```
**** Loop Back Test Failure #1.
```

After the write operation is complete, the FEP/OS is notified that data is available for transmission. At this time, the 128 bytes of data is transmitted. If the transmission fails, the following message will appear:

```
**** Loop Back Failure Sending 128 Bytes from Buffer.
```

Probable cause of failure:
- Hardware failure.
- FEP/OS or driver failure.

Phase 3 At this point, the FEP/OS should have received 128 bytes of data. If not, the following message will appear:

```
**** Loop Back Failure Receiving 128 Bytes.
**** Data Not Transferred to RX.
```

Probable cause of failure:
- Hardware failure.
- Loop back connector not installed, or incorrectly wired.
- FEP/OS or driver failure.

Phase 4 128 bytes of data are read from the receive buffer. If the read operation fails, the following message will appear:

```
**** Loop Back Failure Reading 128 Bytes from Buffer.
```

Probable cause of failure:
- Hardware failure.
- FEP/OS or driver failure.

Phase 5 The data that was transmitted is compared with the data that has been read from the receive buffer. If the comparison fails, the following message will appear:

```
**** Loop Back Failure Verifying 128 Bytes.
**** Data Incorrectly Transferred
```

Probable cause of failure:

- Hardware failure.
- FEP/OS or driver failure.

If all test phases pass, the following message will appear:

```
Loop Back Test Passed.
```

Press the **Q** key to return to the channel monitoring screen.

Diagnostic Loop Back Plug Specifications

AccelePort

The loopback plug for RJ-45 versions (Digi P/N 60000401) consists of a single 10-pin RJ-45 plug wired as follows:

Pin 3 connected to pins 1 and 8 (RTS to RI and CTS)
Pin 5 connected to pin 6 (TxD to RxD)
Pin 9 connected to pins 2 and 10 (DTR to DSR and DCD)

For EIA-232 DB-25 versions, the loopback plug (Digi P/N 60000442) consists of a female DB-25 connector wired as follows:

Pin 4 connected to pins 22 and 5 (RTS to RI and CTS)
Pin 2 connected to pin 3 (TxD to RxD)
Pin 20 connected to pins 6 and 8 (DTR to DSR and DCD)

For EIA-422 DB-25 versions, the loopback plug (Digi P/N 60000638-01) consists of a female DB-25 connector wired as follows:

Pin 2 connected to pin 3 (TxD+ to RxD+)
Pin 4 connected to pin 5 (RTS+ to CTS+)
Pin 14 connected to pin 16 (TxD- to RxD-)
Pin 19 connected to pin 13 (RTS- to CTS-)

Avanstar

The loopback plug (Digi P/N 799001-03) consists of a male DB-25 connector wired as follows:

Pin 4 connected to pins 5 and 8 (RTS to CTS and DCD)
Pin 2 connected to pin 3 (TxD to RxD)
Pin 20 connected to pins 6 and 22 (DTR to DSR and RI)
Pin 11 connected to pin 13 (EIA-422 TxD+ to RxD+)
Pin 14 connected to pin 16 (EIA-422 TxD- to RxD-)

Uninstalling the MPI Software Package

Uninstalling the MPI software package is a two-part procedure. First, you must remove any device drivers that were installed via the MPI software, then you remove the MPI software package itself.

Removing Device Drivers

1. To run the MenuPort Interface program, log in as root and enter:

```
mpi
```

2. From the Main menu, select **Config**.

3. Now select **Remove**, from the Config menu.

4. From the Pick List, select the device driver you wish to remove. Repeat this step for all installed device drivers.

Removing the MPI Software Package

SCO UNIX

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 `mpi`, then select `all`, to remove the entire package.

```
Removing Digi MenuPort Interface Package.
```

4. Quit `custom`, and perform a system shutdown and reboot to activate the new kernel.

System V Release 4

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

2. Enter the following command:

```
pkgrm -n mpi
```

Setting Terminal Options with ditty

/usr/bin/ditty is a utility program that sets and displays the terminal options for Digi intelligent serial products.

The format is:

ditty [**-a**] [**-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.

Note:

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** if you are running SCO UNIX, or **/etc/rc2.d** if you are running System V, Release 3 or 4—see your System Administrator's Guide for details). Alternatively, you may include the **ditty** command sequence in your **.login** or **.profile** files, to ensure that the options you want are activated when you log in. Your system administrator can help you edit these files.

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**.

-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** (SCO) or **term/***xxx* (SVR4). 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.

clear Clear the **tbusy** flag in the channel structure.

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.

stopout Stop output exactly as if an **xoff** character was received.

startout Restart stopped output exactly as if an **xon** character was received.

stopin Activate flow control to stop input.

startin Release flow control to resume stopped input.

[-]dtr Raise [drop] the DTR modem control line, unless DTR hardware flow control is selected.

[-]rts 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.

[-]printer Forces port to stay open so that non-sticky parameters such as baud rate and flow control are not reset to default values.

[-]fastcook Perform cooked output processing on the intelligent card to reduce host CPU usage, and increase raw mode input performance. *This option is not available in System V Release 4.*

`[-]fastbaud` Alter the baud rate tables by the following mapping scheme.

	Xe	***Xr, Xem, EPC/X***	***Xr 920***	***C/X***	***Avanstar***
Specified Baud Rate:	**Baud Rate Mapped to:**	**Baud Rate Mapped to:**	**Baud Rate Mapped to:**	**Baud Rate Mapped to:**	**Baud Rate Mapped to:**
50	57600	57600	57600	57600	57600
75	76800	76800	76800	75	76800
110	115200	115200	115200	115200	115200
134	14400	14400	14400	14400	14400
150	28800	57600	57600	57600	57600
200	200	230400	230400	200	200
300	300	76800	76800	300	76800
600	600	115200	115200	115200	115200
1200	1200	230400	230400	1200	1200
1800	1800	28800	28800	28800	28800
2400	2400	2400	460800	2400	2400
4800	4800	4800	921600	4800	4800
9600	9600	9600	9600	9600	9600
19200	19200	19200	19200	19200	19200
38400	38400	38400	38400	38400	38400

[-]rtspace Enable [disable] RTS hardware input flow control, so RTS drops to pause remote transmission.

[-]ctspace Enable [disable] CTS hardware output flow control, so local transmission pauses when CTS drops.

[-]dsrpace Enable [disable] DSR hardware output flow control, so local transmission pauses when DSR drops.

[-]dcdpace Enable [disable] DCD hardware output flow control, so local transmission pauses when DCD drops.

[-]dtrpace Enable [disable] DTR hardware input flow control, so DTR drops to pause remote transmission.

[-]forcedcd Disable [re-enable] carrier sense, so the tty may be opened and used even when carrier is not present.

[-]altpin Switches the function of the DSR and the DCD inputs on the interface connector, so that DCD is available when using an 8-pin RJ-45 connector instead of a 10-pin RJ-45 connector.

eia232|eia422

By default, each port on an Avanstar configurable connection panel powers up in EIA-232 mode. This option sets the EIA mode for the given port to either EIA-232 or EIA-422. *This option is available only for Avanstar adapters with configurable connection panels.*

startc *c* 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).

stopc *c* Sets the XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see **startc**, above, for format of octal and hexadecimal numbers).

astartc *c* Sets auxiliary XON flow control character. The character may be given as a decimal, octal, or hexadecimal number (see **startc**, above, for format of octal and hexadecimal numbers). *This option is not available in System V Release 4.*

astopc *c* Sets auxiliary XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see **startc**, above, for format of octal and hexadecimal numbers). *This option is not available in System V Release 4.*

[-]aixon 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. *This option is not available in System V Release 4.*

[-]2200flow Use 2200 style flow control on the port. The 2200 terminals support an attached printer and use four flow control characters: terminal XON (0xF8), printer XON (0xF9), terminal XOFF (0xFA) and printer XOFF (0xFB). *This option is not available in System V Release 4.*

[-]2200print

The **2200print** flag determines how these flow control characters (see **2200flow**, above) are interpreted. If **2200print** is set, run independent flow control for terminal and transparent print devices. Otherwise, terminal and printer flow control are logically tied together, so if either XOFF character is received, all output is paused until the matching XON character is received. *This option is not available in System V Release 4.*

edelay *n* **edelay** is a tunable parameter used to set the number of milliseconds of delay between the time the first character arrives after a period of no characters and notification of its arrival to the host. This is also referred to as the wakeup rate between the host adapter software (FEPOS) and the host device driver. This has the advantage of reducing host overhead by allowing the host to process larger blocks of incoming data.

Larger **edelay** values result in more characters being sent in a given time period. This will reduce host processor utilization and character response time and increase overall system throughput.

Smaller **edelay** values result in fewer characters being sent in a given time period. This will increase character response time and increase host processor utilization.

The default value for **edelay** is 100. This is a good value for normal tty activity including typing and *uucp*. For applications receiving continuous input at high speeds, increasing **edelay** will result in lowering host overhead and an increase in overall system throughput. A value of 250 is reasonable.

maxcps *n* 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.

maxchar *n* 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.

bufsize *n* 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 **maxcps** rate selected above. Default is 100 characters.

onstr *s* Sets the terminal escape sequence to turn transparent printing on. The string *s* 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), *s* should be entered as "**\033[5i**".

offstr *s* Sets the terminal escape sequence to turn transparent printing off. See "**onstr**", above, for the format of the string *s*.

term *t* 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**, **vt320**, **vt420**, **wang2x36**, **wyse30**, **wyse50**, **wyse60**, or **wyse75**. If the terminal type is not found in the internal default table, then **ditty** reads the **terminfo** 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)**.

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 27 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`** if you are running SCO UNIX, or **`/etc/rc2.d`** if you are running System V, Release 3 or 4—see your System Administrator's 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)*] *port*

The command line options are:

maxcps *n* Sets the maximum Characters Per Second (CPS) rate at which characters are output to the transparent print device. See *Setting Terminal Options with ditty* (page 27) for more information.

maxchar *n* Sets the maximum number of transparent print characters the driver will place in the output queue. See *Setting Terminal Options with ditty* for more information.

bufsize *n* Sets the driver's estimate of the size of the transparent printer's input buffer. See *Setting Terminal Options with ditty* for more information.

onstr *s* Sets the terminal escape sequence to turn the transparent printer on. An arbitrary octal character *xxx* may be given as ***\xxx***.

offstr *s* Sets the terminal escape sequence to turn the transparent printer off. An arbitrary octal character *xxx* may be given as ***\xxx***.

term *t* Specifies the terminal type. See *Setting Terminal Options with ditty* for more information.

port Specifies the tty device (e.g. **ttyi1a**).

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

Example: **cat** *filename* **> /dev/pri1a**

ditty Examples for Transparent Printing

Example 1:

The following command configures the DigiPRINT options for a DEC VT100 terminal connected to **/dev/ttyi1a**. (Note that the printer will be called **/dev/pri1a**). **maxcps**, **maxchar** and **bufsize** are left to defaults. Enter:

```
ditty term vt100 /dev/ttyi1a
```

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. Enter:

```
ditty onstr "\033[5i" offstr "\033[4i" /dev/ttyi1a
```

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. Enter:

```
ditty term wyse30 maxcps 75 maxchar 100 bufsize 1000 /dev/ttyi1a
```