RealPort on Linux

Setup Guide

92000316A

Contents

chapter 1

Introduction

In This Chapter

About This Guide

Purpose

Use this guide for the following:

- An introduction to RealPort, how to install, configure and use RealPort and, how to troubleshoot should you encounter a problem
- Configuration examples
- Administration procedures

Audience

This manual is intended for the person responsible for configuring and administering RealPort. This person should have experience configuring network devices and be familiar with networking concepts.

Conventions

Following are the conventions used in this document:

Special Fonts

A special font is used for any input you need to enter. For example,

```
set config
```

Square Brackets

Optional parameters are displayed within square brackets. For example,

```
set config [dhcp=on]
```

Note: The square brackets themselves are not actually part of the command, and should not be entered.

Italics

Variables are displayed in italics. For example,

```
set config ip=ip-address
```

Note: Substitute an appropriate IP address for *ip-address* in the above command.

Vertical Bar

A vertical bar character (|) is used to denote a choice (logical "or"). For example,
`set flow=on|off`

Note: The above command would be entered as either:
`set flow=on` or `set flow=off`

About the RealPort Driver

The RealPort driver controls communication between the RealPort driver module and your Digi devices across the network. One driver must remain in the background for each Digi device, maintaining the communication link. You should not need to start the driver by hand. The dgrp_cfg_node program does this automatically. See Managing a Digi Device with Digi RealPort Manager, on page 3-9, or Managing a Digi Device with dgrp_cfg_node, on page 3-12.

To determine whether any RealPort drivers are currently running on your system, as well as to determine which Digi device each driver is communicating with, you can execute the following command:

```
ps ax | grep drpd
```

For more information about the RealPort driver, see the drpd man page which is installed with the driver package.

About the Driver Module

The Linux RealPort driver package consists of a number of utilities in support of two core components; dgrp.o, the RealPort driver module and drpd, the RealPort driver. For more information on the RealPort driver, see the previous topic.

The dgrp.o driver is a standard Linux module and can be manipulated with the standard tools such as lsmod, insmod, rmmod, etc. It should be unnecessary to execute any of these commands since most operations can be performed with the dgrp_cfg_node program. See Managing a Digi Device with dgrp_cfg_node, on page 3-12.

Use this command to determine whether the module is currently loaded:

```
lsmod | grep dgrp
```

One effect of the loading of the driver module is the creation of a directory tree in the /proc file system. Files and directories under the /proc/dgrp directory are used by the RealPort tools to maintain the driver. One file in particular, /proc/dgrp/config, can be examined to determine the current configuration and state of the Digi devices attached to the driver.

For more information about the driver module, see the dgrp man page which is installed with the driver package.

chapter 2

Planning and Requirements

In This Chapter

Requirements and Considerations

Read this section before beginning the RealPort driver installation.

- At the time this document was created, the Linux RealPort driver supported the 2.0, 2.2, and 2.4 kernels. To determine if a particular version of the Linux RealPort driver supports a particular version of Linux, check the release notes for that driver.
- At the time this document was created, the RealPort software **did not** support transparent printing, despite the fact that transparent print devices are created when the driver is installed. To determine if transparent printing is supported by a subsequent version of the Linux RealPort driver, check the release notes for that driver.
- RealPort utilities are located in /usr/bin/dgrp.
- Once the RealPort driver is installed, man pages are available for a number of the utilities associated with the package, including:

`ditty-rp`	which sets and displays RealPort TTY device options
`dgrp_cfg_node`	a tool used to add and remove Digi devices from the RealPort driver
`drpd`	the Digi RealPort network daemon
`dgrp.o`	the RealPort driver module. These are placed in /usr/bin/dgrp. The actual man page entry associated with the drgp.o module is dgrp.
`dgrp_gui`	the Digi RealPort Manager for x-windows

Information to Gather

Before you install RealPort, you need to determine:

- The hostname or IP address to assign each unconfigured Digi device.
- The number of ports for each Digi device. Include any Xem PORTS Module ports that are attached to the Digi device.
- Determine the link speed between the host machine and the Digi device if a slow WAN link (for example, a 56K leased line) connects them.

Configuring a Digi One RealPort, Digi One IA RealPort, or PortServer for RealPort

Use this procedure to configure a Digi One RealPort, Digi One IA RealPort, or PortServer for use with Digi RealPort drivers.

Note: See the appropriate Command Reference or Configuration and Administration Guide for information on how to perform the steps below.

Procedure

1. Access a root prompt on the Digi device.
2. Set the Digi device's IP address.
3. Verify that the RealPort TCP port number is set to 771. Change if necessary.
4. For all ports that will be using RealPort, set the device type to rp. The following example configures ports 2 through 16 of a PortServer for RealPort:

```
set ports range=2-16 dev=rp
```

Configuring EtherLite for RealPort

Use this procedure to install and configure an EtherLite Terminal Server for use with Digi RealPort drivers.

Procedure

1. Attach the EtherLite module to the network using an appropriate cable.
2. Assign an IP address to the EtherLite module (see the following topic).

Assigning an IP Address

EtherLite IP addresses may be set by either of two methods:

- dgipserv, a UNIX utility for setting EtherLite IP addresses (see below)
- EtherLite Boot Console (see the EtherLite Administration Card)

Using dgipserv to Assign an IP Address

dgipserv is a Digi utility used to set the IP addresses for Digi devices. It will also set the gateway and subnet mask addresses when needed to upgrade the Digi device's firmware. For more information on dgipserv consult the man pages.

Note: dgipserv will only work with firmware version 7.9 or later.

Procedure

1. Enter the command:

   ```
   dgipserv
   ```

2. To store an IP addresses in your Digi device enter:

   ```
   dgipserv -store MAC_address(IP_address)
   ```

   where the MAC address and the IP address are the addresses of the Digi device. An example of this command is:

   ```
   dgipserv -store 10:e0:f7:15:20:8g 143.182.5.63
   ```

Related Documentation

PortServer Documentation

- PortServer Configuration and Reference Guide
- Digi One/PortServer Cable Guide

PortServer II Documentation

- PortServer II Hardware Installation Guide
- PortServer II Command Reference
- PortServer II Configuration and Administration Guide
- Digi One/PortServer Cable Guide
- Digi Port Authority--Remote Device Monitor Setup Guide

PortServer TS 8/16 Documentation

- PortServer TS 8/16 Command Reference
- PortServer TS 8/16 Configuration and Administration Guide
- Digi One/PortServer Cable Guide
- Digi Port Authority--Remote Device Monitor Setup Guide

Digi One/PortServer TS 2/4 Documentation

- Digi One/PortServer TS 2/4 Quick Reference Card
- Digi One/PortServer TS 2/4 Command Reference
- Digi One/PortServer TS 2/4 Configuration and Administration Guide
- Digi One/PortServer Cable Guide
- Digi Port Authority--Remote Device Monitor Setup Guide

Digi One IA RealPort Documentation

- Digi One/PortServer TS 2/4 Command Reference
- Digi One/PortServer TS 2/4 Configuration and Administration Guide
- Digi One/PortServer Cable Guide
- Digi Port Authority--Remote Device Monitor Setup Guide

EtherLite Documentation

- EtherLite Hardware Information Guide
- EtherLite Administration Card
- EtherLite Cable Guide

chapter 3

RealPort Setup

In This Chapter

Master Setup Process

Use this process to install and configure the Linux RealPort driver.

1. Read any release notes that may be included with the installation media.
2. Gather information necessary for installation. See Information to Gather, on page 2-2.
3. Install the RealPort driver. See Installing the Driver: Introduction, below.
4. Configure the PortServer for RealPort. See:
   - Configuring a Digi One RealPort, Digi One IA RealPort, or PortServer for RealPort, on page 2-2
   - Configuring EtherLite for RealPort, on page 2-3
5. Add and configure your Digi One RealPort or PortServers in Linux. There are two methods for doing this. See:
   - Managing a Digi Device with Digi RealPort Manager, on page 3-9
   - Managing a Digi Device with dgrp_cfg_node, on page 3-12

   Note: Digi RealPort Manager uses a graphical interface, while dgrp_cfg_node is command line-based.
6. Configure your ports for operation. See Device Configuration Overview, on page 3-15.

Installing the Driver: Introduction

Digi Supports three distribution methods for the RealPort device driver package.

CD Method

Use this procedure to install a Digi driver from the Digi CD front end menu. This procedure assumes that you have accessed the Digi RealPort CD and that the front end menu appears on your screen. seeInstalling the Driver: CD, on page 3-3.

RPM Method

This method uses a source RPM package. The source RPM method depends on rpm tools. If your Linux distribution does not support RPM packages, or if your system does not have the RPM tools installed, you must either locate and install the RPM manipulation tools or use the TGZ method. To install with this method, see Installing the Driver: RPM Methods, on page 3-3.

Note: Digi does not support the distribution of the RealPort software via binary RPMs.

TGZ Method

This method uses a compressed archive of the source. The TGZ method, while relatively simple, does not have the advantage of the common package management operations of RPM-based packages. It does, however, have the advantage that far more systems will support the TGZ method “out-of-the-box”. To install with this method, seeInstalling the Driver: TGZ Method, on page 3-6.

Installing the Driver: CD

Use this procedure to install software or a driver from the Digi CD. You should have already started the Digi CD front-end program.

Procedure

1. Choose an operating system.
2. Choose the Hardware product name.
3. Choose the software or driver you wish to install.
4. Choose Install Software.
5. Follow the prompts.
6. Reboot your system if necessary.

Installing the Driver: RPM Methods

Use one of the following procedures to install the Linux RealPort driver.

Note: The procedures in this topic should be used only in Linux environments that support the installation and building of packages from an RPM repository. The procedures require that the following RPM directories exist and are used by RPM on your Linux system.

- /usr/src/redhat/BUILD
- /usr/src/redhat/RPMS
- /usr/src/redhat/RPMS
- /usr/src/redhat/SOURCES
- /usr/src/redhat/SPECS
- /usr/src/redhat/SRPMS

Installation Methods

There are three methods of installing with RPM:

Method	Description
Installing with digirpm	digirpm is a shell script designed to automate the build and installation process from a source RPM. This installation is fully automated.
Standard Installation Procedure	This method is still automated but there is user control at every step.
Custom Installation Procedure	This method is intended for use by users who wish to customize the installation (for example, to change the destination directories of certain tools). This method is the most complex and should be used only if necessary.

All three methods are discussed in detail on the pages that follow.

Package Version and Revision

Certain commands used in the following procedures need to be entered with the correct package version and revision number.

Note: To determine package version and revision numbers, use this command:

```
rpm -qp filename
```

Example:

If the package version of the Linux driver is 1.0 and the revision is 2, then:

For this command variable:	This value would be used:
version	1.0
revision	2

For this command:	This would be entered:
`rpm -ivv realport-`*`version revision`*`.src.rpm`	`rpm -ivv realport-1.0-2.src.rpm`

Installing with digirpm

Use this procedure to install the RealPort Linux driver with digirpm.

1. Download the RPM package as well as the digirpm tool. The RPM package and digirpm tool can be found on Digi's web site, www.digi.com, or Digi's ftp site, ftp.digi.com/support/drivers/linux.
2. Copy the RPM file and digirpm to a directory on the Linux system. Create a new directory for the two files if you wish.
3. In the same directory as the RPM and digirpm files, execute the following command to extract the source from the source RPM package, build a binary RPM native to your system, and to install the package:

   ```
   sh digirpm filename
   ```

   where filename is the name of the driver file.

   Note: For more information about how to use the digirpm tool, use this command:

   ```
   sh digirpm -h
   ```

Standard Installation Procedure

Use this procedure to install the RealPort Linux driver without digirpm. See Package Version and Revision, above, for important information on entering commands in this procedure.

1. Download the source RPM package. The RPM package can be found on Digi's web site, www.digi.com, or Digi's ftp site, ftp.digi.com/support/drivers/linux. For consistency, you might consider downloading the source RPM to the directory /usr/src/redhat/SRPMS.
2. Install the source code in the SOURCES directory used by RPM (/usr/src/redhat/SOURCES) with this command:

   ```
   rpm -ivv realport-version-revision.src.rpm
   ```

   This command also copies the specification file (/usr/src/redhat/SPECS/realport-v.v.spec) to the SPECS directory.
3. Create an RPM specific to your platform by executing the following commands:

   ```
   cd /usr/src/redhat/SPECS
   rpm -bb realport-version.spec
   ```

4. Install the RPM which was just created by the previous step using the following commands.

```
cd /usr/src/redhat/RPMS/arch
rpm -ivv realport-version-revision.arch.rpm
```

The *arch* value should be replaced with a string representing your architecture, such as i386 or alpha. This value is system dependent, and is chosen by your system when the RPM is built.

Note: To install the RealPort driver on another system of the same architecture and Linux version, copy the following binary to the same location on the other system and execute the rpm command in this step:

```
/usr/src/redhat/RPMS/arch/realport-version-revision.arch.rpm
```

5. Use the following commands to verify that all files are installed correctly:

```
cd /usr/src/redhat/SPECS
rpm -bl realport-version.spec
```

Custom Installation Procedure

Important: This custom installation procedure is reserved for those users that need to change a portion of the installation procedure or its contents in their specific environment. Reasons might include a need to change the directory where files are installed, or to modify the source code. **Modifications to the source package may cause Digi to refuse support for that package in that environment**.

See Package Version and Revision, on page 3-4, for important information on entering commands in this procedure.

1. Download the RPM package. The RPM package can be found on Digi's web site, www.digi.com, or Digi's ftp site, ftp.digi.com/support/drivers/linux. For consistency, you might consider downloading the source RPM to the directory /usr/src/redhat/SRPMS.
2. Install the source code in the SOURCES directory used by RPM (/usr/src/redhat/SOURCES) with this command:

```
rpm -ivv realport-version-revision.src.rpm
```

This command also copies the specification file (/usr/src/redhat/SPECS/realport-*version*.spec) to the SPECS directory.

3. Use the RPM tools to open the source archive:

```
cd /usr/src/redhat/SPECS
rpm -bp realport-version.spec
```

The -bp option specifies that only the preparation section (%prep) of the specification file should be executed. This might result in the source files being uncompressed, removed from the archive, and placed in the following directory:

```
/usr/src/redhat/BUILD/realport-version
```

4. Make desired modifications to the source files and/or the specification file. The source files are found in the directory

```
/usr/src/redhat/BUILD/realport-version
```

The specification file is named:

```
/usr/src/redhat/SPECS/realport-version.spec
```

Important: You should save the original version of any file that you change, plus the final version of the files after you make and test your changes. These backup copies should be placed somewhere other than in the /usr/src/redhat tree.

Important: If you change the final location of any of the files in the package, you must also update the %file list in the specification file. Otherwise, you will get a message that the installation was incomplete, because some files were not found.

Recommended: It is recommended that you document your changes in the %description section of the specification file.

5. The easiest method for building and installing the modified package is to first replace the Digi TGZ archive with a new one including your changes. Replace the archive by doing the following:
   a. Change to the appropriate directory:

      ```
      cd /usr/src/redhat/BUILD
      ```

   b. Create a compressed archive of the source files: (note that the following command should be entered on a single line)

      ```
      tar -cvzf ../SOURCES/realport-version.tgz realport-version/*
      ```

Once the TGZ file has been replaced, you can proceed with step 3 of the Standard Installation Procedure, on page 3-4.

Installing the Driver: TGZ Method

The following procedure can be used in any Linux environment. It is the only choice available for environments which do not support RPM.

Package Version and Revision

Certain commands used in the following procedure need to be entered with the correct package version and revision number.

Note: Refer to release notes for version and revision (release) numbers.

For this command:	This would be entered:
`tar -xvzf realport-version revision.tgz`	`tar -xzvf realport-1.0-2.tgz`

Procedure

1. Download the Linux RealPort TGZ file. This file can be found on Digi's web site, www.digi.com, or Digi's ftp site, ftp.digi.com/support/drivers/linux.
2. Choose a directory (such as /usr/src) where the source tree will reside, and unpack the compressed archive file there. For example:

   ```
   cd /usr/src
   tar -xvzf realport-version-revision.tgz
   ```

   This will create a subdirectory called realport-*version* containing all of the RealPort source files.
3. Change directory to the root of the source directory tree.

   ```
   cd /usr/src/realport-version
   ```

4. Examine the Makefile and make modifications as required by your system environment. Common items to check would include destinations directories, naming conventions, and compiler details.
5. When satisfied with the state of the Makefile and other source files, compile the driver and its support tools by entering this command:

   ```
   make all
   ```

6. Install the package components with:

   ```
   make install
   ```

7. Register your module with the system initialization scripts by entering:

   ```
   make postinstall
   ```

Recommended: Do not delete the source tree since the makefiles are necessary for the uninstall procedure.

Uninstalling the Driver

Use one of the following methods to uninstall the Linux driver.

Uninstalling with RPM

Note: The procedures in this topic should be used only in Linux environments which support the installation and building of packages from an RPM repository. The procedures require that the following RPM directories exist and are used by RPM on your Linux system:

- /usr/src/redhat/BUILD
- /usr/src/redhat/RPMS
- /usr/src/redhat/SOURCES
- /usr/src/redhat/SPECS
- /usr/src/redhat/SRPMS

Procedure

1. Use the dgrp_cfg_node command with the uninit option for each PortServer. This will kill each PortServer daemon and erase all information of that PortServer from the system.

   ```
   dgrp_cfg_node uninit ID
   ```

   where ID specifies the ID of the PortServer to unconfigure.

2. Enter this command to remove the driver package:

   ```
   rpm -e -vv realport
   ```

Uninstalling from a TGZ Archive

Important: If you have deleted the source repository created when you installed this driver package, then there is no automated mechanism to remove the package files.

1. Access a command prompt at the root on your Linux system.
2. Change directories to the root of the source tree

   ```
   cd /usr/src/realport-version
   ```

   Substitute your driver version numbers for *version* in the previous command. For example, if the driver version is 1.0, the command would be:

   ```
   cd /usr/src/realport-1.0
   ```

3. Enter these two make commands:

   ```
   make preuninstall
   make uninstall
   ```

RealPort Devices

When installed, the Linux RealPort driver creates three different devices for each physical port of each Digi device: a standard TTY device, a callout device, and a transparent print device.

Device Naming Convention

The devices are named according to the following convention:

[*prefix*][*ID*][*port*]

These elements are defined as follows:

Element	Description
prefix	Standard TTY devices have the prefix “tty”, callout devices have the prefix “cu”, and transparent print devices have the prefix “pr”.
ID	The RealPort ID for the Digi device associated with this port. A RealPort ID consists of one or two alphanumeric characters. An underscore character may be used for any of the two ID characters.
port	The port number must consist of two digits. The ports are numbered beginning with 00.

The following are examples of the devices which would be created for the first port of a Digi device with the RealPort ID “aa”.

Device Type	Full Path Name
Standard TTY Device	/dev/ttyaa00
Callout Device	/dev/cuaa00
Transparent Print Device	/dev/praa00

Standard TTY Devices

The behavior of the standard tty devices is that of a modem controlled port. They require the Data Carrier Detect (DCD) signal to be high before they will operate. When used on a dial-in modem, the ports will wait for DCD before sending out the login prompt.

When these devices are used with a terminal or other locally connected device, it is usually the practice to wire the DCD signal to the remote equipment's Request To Send (RTS) line. When a terminal is then used for log-in, the system will generate a prompt when the terminal is powered-on (RTS, and thus DCD is asserted), and will kill the user session if the terminal is powered-off (lowering the signals).

Callout Devices

These devices will be obsoleted in a future version of Linux, so should generally be avoided. Data Carrier Detect (DCD) need not be present to open the device.

Once a connection is established and DCD becomes active, these devices behave in the same way as the standard tty devices. Subsequent loss of the DCD signal will cause active processes on the port to be killed.

Transparent Print Devices

The transparent print device can be used with auxiliary printer ports on terminals. Output to the “pr” device goes out the auxiliary port of a terminal while you continue to use the terminal normally.

Important: At the time this document was created, the RealPort software **did not** support transparent printing, despite the fact that transparent print devices are created when the driver is installed. To determine if transparent printing is supported by a subsequent version of the Linux RealPort driver, check the release notes for that driver.

Managing a Digi Device with Digi RealPort Manager

Introduction

Digi RealPort Manager can be used to perform these tasks. .

Task	Procedure
Add a Digi Device	See Adding a Digi Device, on page 3-10.
Delete a Digi Device	See Deleting a Digi Device, on page 3-10.
Start a RealPort daemon	See Starting a RealPort Daemon, on page 3-10.
Stop a RealPort daemon	See Stopping a RealPort Daemon, on page 3-10.
Monitor port status	See Monitoring Port Status, on page 3-11.

The Main Screen

The main screen of the Digi RealPort Manager is used to manage the Digi devices registered with the RealPort driver. It lists all of the currently configured Digi devices in the central window, and provides buttons and menu options to manipulate these Digi Devices.

Menu Field	Description
RealPort ID	The RealPort ID must be one or two alphanumeric characters used to designate your TTY devices for RealPort. This id must be unique to your system. An underscore character may also be used for an ID character. An example for a RealPort device name where the letters rp are used would be: *TTYrp01*
address	The address may be specified as an IP number, or IP name. No attempt is made to validate this address, nor to connect to the specified address during the installation.
ports	This field is an integer indicating the number of device files which should be created for this Digi Device. This does not have to match the physical number of ports, but physical ports beyond the number specified will not be available. The maximum port count allowed is 64.
IP Port	This is the Digi Device IP port number. Normally this should be left at the default.
Access Mode	Access Mode sets the file protection mode for any device files created.
Owner	This field sets the user ID of the file owner for any device files created. The value must be an integer.
Group	This field sets the group ID of the file owner for any device files created. The value must be an integer.
Link Speed	Link Speed sets the link speed string. The default is auto, and can be used under most circumstances. See the drpd(8) man page for information on the custom speed parameters.

Adding a Digi Device

Use this procedure to add a Digi device to your Linux System.

1. Access Digi RealPort Manager by entering this command at a Linux prompt:

   ```
   /usr/bin/dgrp/config/dgrp_gui
   ```

   Note: Digi will only support use of the GUI with the wish interpreter. The wish tcl interpreter needs to be in the Linux "path" before the dgrp_gui command will work.

2. There are two ways to register a new Digi device with the RealPort package. Either press the Add button that appears at the bottom of the main screen, or choose the PortServer > Add New from the menu. In either case, the Digi device Settings window will appear.

3. Specify values for each of the fields displayed (some of the fields have defaults).

4. Press the Commit button and the RealPort Command Logger window will appear and an appropriate dgrp_cfg_node command will be displayed.

5. Press the Run It button to execute the command and finish adding and initializing your Digi device. Watch for any errors in the log window.

Deleting a Digi Device

Use this procedure to delete a Digi device from your Linux System.

1. Access Digi RealPort Manager by entering this command at a Linux prompt:

   ```
   /usr/bin/dgrp/config/dgrp_gui
   ```

   Note: Digi will only support use of the GUI with the wish interpreter. The wish tcl interpreter needs to be in the Linux "path" before the dgrp_gui command will work.

2. Choose the appropriate Digi device in the list box,

3. Press the Delete button that appears on the bottom of the main screen, or choose PortServer > Delete/Uninitialize from the menu. The Digi PortServer Settings window will appear.

4. Confirm that the correct PortServer is being deleted and press the Remove button. The RealPort Command Logger window will appear and an appropriate dgrp_cfg_node command will be displayed.

5. Press the Run It button to execute the command and complete the removal of the Digi device, along with all of its device nodes.

Starting a RealPort Daemon

Use this procedure to start a RealPort daemon for a Digi device.

1. Access Digi RealPort Manager by entering this command at a Linux prompt:

   ```
   /usr/bin/dgrp/config/dgrp_gui
   ```

   Note: Digi will only support use of the GUI with the wish interpreter. The wish tcl interpreter needs to be in the Linux "path" before the dgrp_gui command will work.

2. Choose the appropriate Digi device in the list box.

3. Choose Daemon > Start Daemon from the menu. The Command Logger screen will appear and a dgrp_cfg_node command will display.

4. Press the Run It button to execute the command and attempt to start the daemon. If the daemon is already running, the command will silently exit.

Stopping a RealPort Daemon

Use this procedure to stop a RealPort daemon or Digi device.

1. Access Digi RealPort Manager by entering this command at a Linux prompt:

   ```
   /usr/bin/dgrp/config/dgrp_gui
   ```

   Note: Digi will only support use of the GUI with the wish interpreter. The wish tcl interpreter needs to be in the Linux "path" before the dgrp_gui command will work.

2. Choose the appropriate Digi device in the list box.
3. Choose the Daemon > Stop Daemon from the menu. The Command Logger screen will appear and a dgrp_cfg_node command will display.
4. Press the Run It button to execute the command and attempt to stop the daemon. If the daemon is not running, the command will silently exit.

Monitoring Port Status

Use this procedure to monitor the status of a RealPort port.

Notes

- When monitoring ports, you can change the port monitoring delay by using the sliding scale along the bottom of the Ports window.
- Modem signal values are only correct for open ports. Ports in the **closed** or **waiting** state may not display the correct modem signals.
- When monitoring ports, the Refresh button may be selected at any time to update the port fields.

Procedure

1. Access Digi RealPort Manager by entering this command at a Linux prompt:

   ```
   /usr/bin/dgrp/config/dgrp_gui
   ```

   Note: Digi will only support use of the GUI with the wish interpreter. The wish tcl interpreter needs to be in the Linux "path" before the dgrp_gui command will work.
2. Choose the appropriate Digi device in the list box.
3. Choose the Ports option of the View menu.
4. Choose the appropriate port in the Ports Window.
5. Press either the Modem Status button or select the Modem Status option of the Ports menu. The Port Status window will appear. A signal will show a red box if it is active or a gray box if it is inactive.

Ports Window Information

These are the fields of the Ports Window:

Field	Description
Port	Port is the port number of an individual port. The first port on a Digi One RealPort or PortServer is port number 0.
Status	The value of this field is either open, closed, or waiting. A port moves into the waiting state when there are devices waiting to open the port (either waiting for an event, or for another process to release the port).
Speed	This value is the last known speed (in bps) of the port. If unknown (for instance, before the port is used for the first time), the value 0 is shown.
Description	This field is not yet active and will always show NA

Managing a Digi Device with dgrp_cfg_node

The dgrp_cfg_node tool is a command-line based configuration program designed to automate a number of steps which are required in order to enable the serial ports on Digi devices for general use. The path to the dgrp_cfg_node executable is /usr/bin/dgrp/config.

The dgrp_cfg_node tool has four modes of operation:

Operation	Description
init operation	This operation can be used to add or reinitialize a Digi device in Linux.
uninit operation	This operation removes a Digi device from Linux.
stop operation	This operation stops a RealPort daemon.
start operation	This operation attempts to start a RealPort daemon.

About the RealPort ID

The RealPort tools refer to an individual Digi device with an ID. You must choose the ID to use for each Digi device, and this ID must be unique within your system. The dgrp_cfg_node init operation allows you to assign an ID to a Digi device.

A RealPort ID consists of one or two alphanumeric characters. The underscore character may be used in the ID.

Once an ID is assigned, the device files in the /dev directory which are created by the package will encode the ID into the device name. This allows one, by inspection, to determine which Digi device with which a particular device is associated. For more information on the device naming, see RealPort Devices, on page 3-8.

The init Operation

The init operation has the following effects:

- If necessary, it will attempt to load the driver module.
- It will determine whether a daemon is already running for the specified node. If it is not running, it will start the daemon.
- It will create all necessary device files in the /dev directory. If a device file exists, the ownership and permissions are preserved.

The standard usage requires a command like:

```
dgrp_cfg_node init ID IPaddr ports
```

The parameters are defined as follows:

Parameter	Definition
init	Indicates the operation being requested
ID	Assigns the supplied ID to the Digi device. A RealPort ID consists of one or two alphanumeric characters. The underscore character may be used in the ID.
IPaddr	Either an IP address or an IP name may be assigned to the daemon via this parameter.
ports	This parameter indicates the number of device files which should be created for this particular Digi device. This does not have to match the physical number of ports, but physical ports beyond the ports value specified will be unavailable. The maximum number of ports supported by the tool is 64.

For further details, see the dgrp_cfg_node man page, which is installed with this package.

The uninit Operation

The uninit operation has the following effects:

- It will determine whether a daemon is running for the specified Digi device, and if so, will kill it.
- It will erase all information about this Digi device from its internal database (this distinguishes the behavior from that of stop; please see below).

The standard usage requires a command like:

```
dgrp_cfg_node uninit ID
```

The parameters are defined as follows:

Parameter	Definition
uninit	Indicates the operation being requested
ID	Specifies which Digi device to unconfigure.

Note: The command will fail if any ports are in use. Be sure to kill all applications using the Digi device ports before uninitialization.

Note: Even if you have removed all of the Digi devices you have previously registered with the driver, the dgrp_cfg_node utility will not unload the driver module.

For further details, see the dgrp_cfg_node man page, which is installed with this package.

The stop operation

The stop operation has the following effect:

- It will determine whether a daemon is running for the specified Digi device, and if so, will kill it.
- The RealPort software retains the information associated with this ID, so that the daemon can be restarted with a minimum of information.

The simplest usage requires one to execute a command with a form similar to:

```
dgrp_cfg_node stop ID
```

The parameters are defined as follows:

Parameter	Definition
stop	Indicates the operation being requested
ID	Specifies which Digi device's daemon to stop.

One might stop a daemon in order to change the daemon's parameters, like the IP address (i.e. stop, then restart with new parameters). For further details, please see the dgrp_cfg_node man page, which is installed with this package.

The Start Operation

The start operation determines whether a daemon is running for the specified node, and if not, will attempt to execute a daemon based on the supplied parameters.

The simplest usage requires one to execute a command with a form similar to:

```
dgrp_cfg_node start ID IPaddr
```

The parameters are defined as follows:

Parameter	Definition
start	Indicates the operation being requested
ID	Specifies which Digi device's daemon to start.
IPaddr	The (possibly new) IP address or IP name to use when referring to the Digi device with the specified ID.

For further details, see the dgrp_cfg_node man page, which is installed with this package.

Device Configuration Overview

Use these overviews of device configuration topics to decide which tool(s) you should use to configure your devices.

ditty-rp

Features or Description	• Digi's device configuration program • Run from a command prompt • ditty-rp commands are normally included in a system startup file.
Uses	• Manually configure Digi-specific device settings such as altpin or forcedcd • Configure printer devices. See Configuring a Device for a Printer, on page 3-16
Resources	• Refer to the man page for ditty-rp • See

Configuring a Port for a:

Terminal	See Configuring a Device for a Terminal, on page 3-16. This procedure describes how to configure a device for a terminal.
Printer	See Configuring a Device for a Printer, on page 3-16. This procedure describes how to configure a device for a printer.
Modem	See Configuring a Device for a Modem, on page 3-17 This procedure describes how to configure a device for a modem.

Miscellaneous Topics

altpin	This option should be enabled on a port when an RJ-45 8-pin cable is used and the Data Carrier Detect (DCD) signal is required. For example, altpin should be enabled on a port where an RJ-45 8-pin cable is used with a modem. See Setting TTY Options, on page 3-18, and refer to the Cable Guide for your Digi product.
fastbaud	RealPort devices support baud rates in excess of the maximum baud rate supported by Linux. To enable the use of these fast baud rates, a ditty-rp parameter, fastbaud, has been provided. See the topic *fastbaud Data Rate Mapping* in Setting TTY Options, on page 3-18.
Data Carrier Detect (DCD)	In some cases, depending on your operating system requirements and/or your device requirements, it may be necessary for the Data Carrier Detect (DCD) signal to be active on a port before it will function. There are two ways to fulfill this requirement. ***Cabling*** One way to fulfill the DCD requirement is to create a cable and have your device force the signal high. See the Digi One RealPort and PortServer Cable Guide. ***ditty-rp*** Another way to fulfill the DCD requirement is to enable the forcedcd ditty-rp option. See Setting TTY Options, on page 3-18.

Configuring a Device for a Terminal

Use this procedure to configure a Digi serial device for a terminal. See your operating system documentation for more information on configuring a serial device for a terminal.

Procedure

1. Connect a serial cable between the port and terminal.
2. Edit the /etc/inittab file and add a getty entry for the device. The getty name that configures your device can vary from system to system. The following examples are only meant to serve as a guide. Substitute your device for ttyaa011 in these examples:

   Example 1: RedHat getty_ps

   An example of a common RedHat terminal install script where getty is getty_ps would be:

   ```
   dl:2345:respawn:/sbin/getty_ps ttyaa11 DT9600 vt100
   ```

   Example 2: Debian agetty

   An example of a common Debian terminal install script where getty is agetty would be:

   ```
   D1:23:respawn:sbin/agetty -L ttyaa11 19200 vt100
   ```

   Example 3: mgetty

   An example of an mgetty modem script which is available both in RedHat and Debian is:

   ```
   T3:23:respawn: /sbin/mgetty -x0 -s 115200 ttyaa11
   ```

3. Enable the port for login by rebooting the system or by entering this command at your Linux command prompt:

   ```
   init q
   ```

Configuring a Device for a Printer

You have two options in configuring your printers with Linux drivers.

1. Connect a serial cable between the port and printer.
2. Use either the Linux printtool command to configure your printer or, set up the printers manually. To install a printer manually, add lines similar to this example to the /etc/inittab file:

   ```
   DG01:2345:once: cat < /dev/ttyaa11 > /dev/null &
   dg01:2345:once: ditty-rp 38400 ctspace altpin -ixon -ixoff -ixany
   /dev/ttyaa11
   ```

   The device /dev/ttyaa11 is used as an example here. Substitute your particular device name in the previous commands. This example also sets the port speed to 38400, enables altpin, enables hardware flow control (ctspace and rtspace) and disables software flow control (-ixon, -ixoff, -ixany) on the port. Configure the ditty-rp parameters as required by your specific printer and cable configuration. For more information on ditty-rp, see Setting TTY Options, on page 3-18.

Configuring a Device for a Modem

Use this procedure to configure a Digi device for a Dial-in/Dial-out modem connection. Configuring a device for a modem requires familiarity with both the operating system and the modem being used. While the following procedure is sufficient for most cases, it may be necessary to take additional steps to properly configure your modem or to set up the operating system for a specific application.

Procedure

1. Connect a serial cable between the port and modem.
2. Power the modem on.
3. At the command prompt enter:

   ```
   chown uucp:uucp /dev/ttyaa00
   ```

   where *ttyaa00* is the name of the device.
4. Connect to the modem by entering this command at a command prompt

   ```
   cu -l /dev/ttyaa00 -s 38400
   ```

   where *ttyaa00* is the name of the non modem control device for the port.
5. Set the modem to answer after the first ring with this command:

   ```
   ats0=1
   ```
6. Train the modem to the port speed with this command:

   ```
   at&w
   ```
7. Enter any other desired modem commands.
8. Terminate the connection to the modem with a tilde followed by a period:

   ```
   ~.
   ```
9. To manually configure the port to use hardware flow control by inserting this command in a Linux startup file:

   ```
   ditty-rp rtspace ctspace /device
   ```

   where /device is the name of the Digi device. Insert the command in a startup file so it remains in effect after a reboot.

   Alternately, use a gettydef entry that uses hardware flow control (see the next step)
10. Edit the /etc/inittab file and add a getty entry for the device. The getty name that configures your device can vary from system to system. The following example is only meant to serve as a guide.

    Example:

    An example of an mgetty modem script which is available both in RedHat and Debian is:

    ```
    T3:23:respawn: /sbin/mgetty -x0 -s 115200 ttyaa00
    ```

    where *ttyaa00* is the name of the device.
11. Enable the port for login by rebooting the system or by entering this command at your Linux command prompt:

    ```
    init q
    ```

Setting TTY Options

The RealPort Linux device driver package includes a command, ditty-rp, which is a superset of stty, and may be used to set and display the device options for Digi RealPort devices.

The general command format is:

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

With no options, ditty-rp 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.

ditty-rp commands may be executed from the command line, or placed in a startup script to be run whenever the system is booted.

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 specified TTY device. Replace *ttyname* with the TTY pathname (such as /dev/ttya01s, /dev/term/a01 or /dev/dty/a001s, depending on your operating system). 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 cancelled 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 were received.
startout	Restart stopped output exactly as if an XON character were 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.

[-]fastbaud	Alter the baud rate tables to permit the use of data rates that are beyond the range supported by the operating system. See fastbaud Data Rate Mapping, on page 3-20.
[-]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.
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).
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).
[-]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.
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*" — Defines the terminal escape sequence to direct subsequent data to the transparent printer.

s is a string of ASCII characters, enclosed in quotes, that command the terminal to enter transparent printing mode. An arbitrary octal character xxx may be given as \xxx.

For example, the sequence <Esc>[5i would be entered as: `"\033[5i"`.

offstr "*s*" — Defines the terminal escape sequence to stop directing data to the printer.

s is a string of ASCII characters, enclosed in quotes, that command the terminal to enter transparent printing mode. An arbitrary octal character xxx may be given as \xxx.

For example, the sequence <Esc>[5i would be entered as: `"\033[5i"`.

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, 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.

fastbaud Data Rate Mapping

Use the table below to see how setting fastbaud affects RealPort data rates.

Specified Data Rate:	Data Rate Mapped to:
50	57600
75	76800
110	115200
134	131657
150	153600
200	230400
300	460800