Digi Connect® Family

Command Reference

Revision history—90000566

Revision	Date	Description
N	September 2013	Updated several of the commands.
P	June 2015	Added product introduction for ConnectPort TS 16 MEI and added new and changed content to accommodate ConnectPort TS 16 MEI.
R	April 2017	Rebranded with minor updates.

Trademarks and copyright

Disclaimers

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, 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.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Send comments

Documentation feedback: To provide feedback on this document, send your comments to techcomm@digi.com.

Customer support

Digi Technical Support: Digi offers multiple technical support plans and service packages to help our customers get the most out of their Digi product. For information on Technical Support plans and pricing, contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Contents

Digi Connect Family Command Reference

Command Descriptions

Modem Emulation Commands

Digi Connect Family Command Reference

This book describes the commands in the command-line interface for the following product families:

- Digi Connect Family Products
- ConnectPort® TS Products
- Digi Cellular Family Products:
- Digi Connect WAN Family
- ConnectPort WAN Family
- ConnectPort X Family Products
- ConnectPort X5 Family Products
- ConnectPort Display
- AnywhereUSB®

This chapter provides the following information:

- Basic information that applies to all commands, including navigation and editing keys, displaying online help, abbreviating commands, syntax conventions, and entering special characters in string values.
- How to access the command line.
- How to configure an IP address for a Digi device from the command line, if an address has not already been assigned.
- Information about user models and user permissions in Digi devices, and how they affect the commands you can issue.

Access the Command Line

To configure devices using commands, you must first access the command line, and then log on with your user name and password.

This procedure assumes that you have already configured the Digi device with an IP address.

1. To access the Command-Line Interface for the Digi device, enter the following command from a command prompt on another networked device, such as a server:

   ```
   #>telnet ip address
   ```

   where *ip address* is the Digi device's IP address. For example:

```
#> telnet 192.3.23.5
```

2. If user authentication has been set up for the device, (that is, a user name and password have been set up for the device), a login prompt is displayed. If you do not know the user name and password for the device, contact the system administrator who configured the device. The default user name is **root** and the default password is **dbps**.

Configure an IP Address

If the device you will be issuing commands has not already been assigned an IP address or the IP address needs to be modified from its initial configuration, see the Digi product's *User Guide* for details on configuring an IP address.

Basic Command Information

Navigation and Editing Keys

Navigation and Editing Keys

Use the keys listed in the table to navigate the command line and edit commands:

Action	Keys
Move the cursor back one space.	Ctrl+b
Move the cursor forward one space.	Ctrl+f
Delete the character to the left of the cursor.	Back space or Ctrl+h
Delete the character under the cursor.	Delete
Scroll back through commands.	Ctrl+p
Scroll forward through commands.	Ctrl+n
Execute the command.	Enter

Displaying Online Help

You can access help for all commands as described in the following table.

For information on...	Type
All commands	**?** (with no additional options)
A specific command	**help** [command] OR [command] **?** Example: **help info** Example: **info ?** Example: **set alarm ?**

Abbreviating Commands

You can abbreviate all commands by typing enough letters to uniquely identify the command.

Syntax Conventions

Presentation of command syntax in this manual follows these conventions:

- Brackets **[ ]** surround optional material.
- Braces **{ }** surround entries that require you to chose one of several options, which are separated by the vertical bar, |.
- Non-italicized text indicates literal values, that is, options or values that must be typed exactly as they appear. Yes and no options are examples of literals.
- Italicized text indicates that a type of information is required in that option. For example, ***filename*** means that the name of a file is required in the option.

Entering Special Characters in String Values

Several commands have options that are string values. For example:

- **set alarm** command - **match** option
- **set autoconnect** command- **connect_on_string** option.

Escape Sequences for Special Characters

You can enter special characters in strings using the following escape sequences:

Escape Sequence	Processed as:
*	Match any character. This escape sequence is only available on the **set alarm match=string** option.
\a	Alert character.
\b	Backspace character.
\f	Form-feed character.
\n	New-line character.
\r	Carriage-return character.
\s	Acts as a separator between characters. This sequence allows entering a string such as **\xB8\s4** where **B8** should be translated as a hexadecimal character separate from the numeric character 4.
\t	Horizontal tab character.
\v	Vertical tab character.
\\	Backslash character (\ ).
\xN	A hexadecimal number, where N is up to 20 hexadecimal digits. For example: **\x10\x2**
\N	An octal byte, where N is up to 3 octal digits. For example: **\2** or **\208**

Length Limitations on String Values

String values for certain command options have specific limitations on the maximum total string value including special characters, and the maximum parsed value (that is, the character-string length when

any escape sequences in the string are processed). The option descriptions note these maximum lengths.

Octal values are limited to a byte (/377). For example, **/377** is translated as octal 377 (equal to \xff), but **/378** is translated as octal 37 (\x1f) then an 8 character.

Commands for managing connections and sessions

Use these commands to manage connections and sessions:

- **close**: Closes active sessions that were opened by **connect**, **rlogin**, and **telnet** commands.
- **connect**: Makes a connection, or establishes a connection, with a serial port.
- **dhcpserver**: Manages DHCP server operation.
- **exit** and **quit**: Terminates a currently active session.
- **vpn**: Manages Virtual Private Network (VPN) connections.
- **who** and **kill**: The who command displays a global list of connections. The list of connections includes those associated with a serial port or the command-line interface. The who command is particularly useful in conjunction with the kill command, which terminates active connections. Use who to determine any connections that are no longer needed, and end the connections by issuing a kill command.
- **ping**: Tests whether a host or other device is active and reachable.
- **reconnect**: Reestablishes a previously established connection; that is, a connection opened by a **connect**, **rlogin**, or **telnet** command; the default operation is to reconnect to the last active session.
- **rlogin**: Performs a login to a remote system.
- **send**: Sends a Telnet control command, such as break, abort output, are you there, escape, or interrupt process, to the last active Telnet session.
- **status**: Displays a list of sessions, or outgoing connections made by **connect**, **rlogin**, or **telnet** commands for a device. Typically, the **status** command is used to determine which of the current sessions to close.
- **telnet**: Makes an outgoing Telnet connection, also known as a session.

User Models and User Permissions in Digi devices

The user model in a Digi device influences the commands that users can issue. There are three user models implemented in Digi devices:

- one-user model
- two-user model
- more than two-user model

Identifying the User Model for Your Digi device

To determine which user model is implemented in your Digi device, issue a **show user** or **set user** command (For more information, see show.).

In the command output, note how many user IDs are defined: one, two, or more than two. You can also issue a **set user ?** command and note the range for the **id** option. If the **id** option is not listed, there is one user. Otherwise, the range for user IDs is displayed.

One-user Model

In the one-user model, by default there is no login prompt, and the default name for user 1 is **root**.

To enable the login prompt, you must issue a **newpass** command with a password length of one or more characters. For more information, see newpass.

Once you have enabled a password is enabled, issuing a **newpass** command with a zero-length password disables it.

- User 1 has a default name of **root**.
- User 1 has permissions that enables it to do all commands. Permissions cannot be altered.

Two-user Model

- User 1 has a default name of **root**. This user is also known as the administrative user.
- User 1 has default permissions that enables it to issue all commands.
- Permissions for User 1 can be changed to be less than the default root permissions.
- User 2 is undefined. That is, it does not exist by default, but it can be defined.
- When defined, User 2 has a limited set of permissions, defined by the **set permissions** command. For more information, see set permissions.
- Permissions for User 2 can be changed to be either greater than or less than its default.

More than Two-user model

User definitions are exactly the same as the two-user model, with the addition of user groups and more users. The **set group** command defines user groups. For more information, see set group.

Login Suppression Feature

You can disable the login prompt by issuing the **set login** command. For more information, see set login.

Increasing Security for Digi Device Users

As needed, you can enforce additional security for device users. For example, you can use the **autoconnect** feature, where a user is automatically connected to another system without accessing the Digi device's command line. For more information, see set autoconnect.

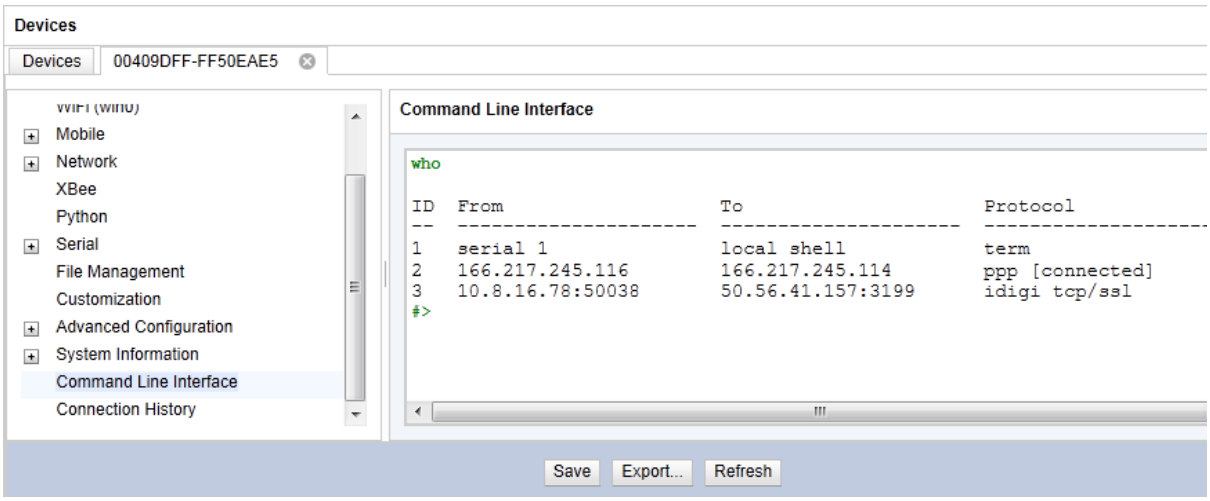
Entering commands from Device Cloud

For Digi devices that are managed by Device Cloud, it is possible to enter the commands documented in this Command Reference within Device Cloud. This allows you to enter commands from the Device Manager interface without having to create a Telnet connection to the Digi device. For example, if you want to view the event log for a particular device from Device Manager, you can enter the **display logging** command directly from that interface.

For more information about managing Digi devices from the Device Cloud, see the *Device Cloud User Guide* and the *Device Cloud Programming Guide.*

To open the command-line interface and enter commands from the Device Cloud Device Manager:

1. In the Device Manager device list, first, navigate to the device details page by doing one of the following:
   - Double-click on the device in the device list.
   - Select one or more devices and then click the **Properties** button in the toolbar.
2. In the menu on the left-hand side of the screen, navigate to **Command Line Interface** toward the bottom of the list.



3. Enter commands following the command syntax as shown in this Command Reference. The example shows entering the **who** command and its output.

Command Descriptions

This chapter provides a description of each command in the command-line interface for the product families included in this manual.

Verify device support for commands

This command reference documents all commands implemented in the firmware for the products listed in the Digi Connect Family Command Reference. Support for commands varies among products. Some products, such as ConnectPort Display and AnywhereUSB, have a limited set of commands, and some commands relate only to particular features unique to specific Digi products. For example, the **set wlan** command applies only to wireless products. Other commands may have options that are specific to features that are not available on all devices. For example, the **display mobile** command applies only to cellular-enabled Digi products.

To verify whether a Digi device supports a particular command or command options, and to get the allowed ranges and limits for command options, you can enter several commands. For example:

- **help** displays all supported commands for a device.
- **?** displays all supported commands for a device.
- **set ?** displays the syntax and options for the **set** command. You can use this to determine whether the device includes a particular **set** command variant.
- **help set** displays syntax and options for the **set** command.
- **set serial ?** displays the syntax and options for the **set serial** command.
- **help set serial** displays the syntax and options for the **set serial** command.

backup

Purpose

Saves the device configuration to a TFTP server located on the network, or restores the configuration from a saved copy on the TFTP server.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions backup=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
backup [to=serveripaddress[:filename]]|
  [from=serveripaddress[:filename]]|
  [print]
  [passwords]
```

Options

to=serveripaddress[:filename]

Stores the configuration file to the specified TFTP server and filename. If a filename is not specified, the command uses the default filename of **config.rci**.

from=**serveripaddress**[**:*filename***]

Restores the configuration file from the specified TFTP server and filename. If a filename is not specified, the command assumes the default filename of **config.rci**.

print

Prints out the current device configuration.

passwords

Includes encrypted passwords and keys.

Example

```
#> backup from=10.0.0.1:config.rci
```

See also

set rciserial: The **set rciserial** command allows a configuration file to be loaded over a serial port when the **DSR** input signal is **asserted** or **raised**.

boot

Purpose

Reboots the Digi device, restores the device configuration to factory default settings, or loads new firmware files (both EOS and POST images) from a TFTP server.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions boot=execute** to use this command. In addition,

- For **action=factory** you also must have **revert-all=execute** permissions.
- For **load=*host ip address*:*load file*** you also must have **fw-update=execute** permissions.

See set permissions for details on setting user permissions for commands.

Syntax

Reboot the device server

```
boot action=reset
```

Restore configuration defaults

```
boot action=factory
```

Load new firmware or POST file into flash ROM from a TFTP host

```
boot load=host ip address:load file
```

Options

action

The action to be performed.

factory

Resets the entire configuration to factory defaults, then reboots the device.

reset

Reboots the device.

load

The firmware to be loaded.

host ip address

The IP address of a host with new firmware or POST file, which is then burned into flash ROM. The host must be running a TFTP server.

load file

The name of a firmware file or POST file. The software automatically detects the type of file and performs the appropriate load operation.

Examples

Restore configuration defaults

This example reloads the firmware stored in flash ROM and resets the configuration to factory defaults then reboots the device.

```
#> boot action=factory
```

Reboot using the current firmware and configuration

This example reboots the device and uses the current firmware and configuration stored in flash ROM.

```
#> boot action=reset
```

Reboot using firmware from a boot host

This example loads the firmware stored on the TFTP host into flash ROM. A reboot is required to use the new firmware.

```
#> boot load=10.0.0.1:firmware.bin
```

See also

revert

certmgmt

Purpose

Displays and manages entries in a database of certificate and private key data. The **certmgmt** supports displaying, loading, saving, removing, certificate database entries, and importing a private key for the Digi device into the database. Certificates and public/private host key pairs are an integral part of public key infrastructure (PKI) based security. The **certmgmt** command manages several kinds of certificate databases and security implementations, including X.509, SSL/TLS, SSH, and VPN.

Note Digi recommends using the web interface instead of **certmgmt** to manage certificate databases and private key data, as it is better suited to the interactive tasks involved. In the web interface, go to **Management > X.509 Certificate/Key Management**.

Tables managed by the "certmgmt" command

Database information is stored in the following tables:

Security type	Table	Used to load
X.509 Certificate Authority/Certificate Revocation	CA (Certificate Authority)	Certificate authority digital certificates. A certificate authority (CA) is a trusted third party which issues digital certificates for use by other parties. Digital certificates issued by the CA contain a public key. The certificate also contains information about the individual or organization to which the public key belongs. A CA verifies digital certificate applicants' credentials. The CA certificate allows verification of digital certificates, and the information contained therein, issued by that CA.
	CRL (Certificate Revocation List)	Certificate revocation lists for loaded CAs. A certificate revocation list (CRL) is a file that contains the serial numbers of digital certificates issued by a CA which have been revoked, and should no longer be trusted. Like CAs, CRLs are a vital part of a public key infrastructure (PKI). The digital certificate of the corresponding CA must be installed before the CRL can be loaded.
Simple Certificate Enrollment Protocol (SCEP)	SCEP CA (Certificate Authority)	SCEP certificate authority digital certificates that have been approved and issued. Tables are populated using SCEP commands and data is obtained from a SCEP server, rather than populated by a user.
	SCEP Pending Enrollment Requests	SCEP certificate requests that are pending approval.

Security type	Table	Used to load
Virtual Private Networking (VPN)	VPN Identity	VPN identity certificates. Identity certificates and keys allow for IPSec authentication and secure key exchange with ISAKMP/IKE using RSA or DSA signatures. The VPN identity certificate must be issued by a CA trusted by the peer.
	VPN Identity Keys	VPN RSA or DSA identity private keys.
Secure Sockets Layer (SSL) and Transport Layer Security (TLS)	SSL Identity	SSL/TLS identity certificates. A default key is generated automatically but can be overridden by a user. However, this default key is not secure.
	SSL Identity Keys	SSL/TLS identity private keys.
	SSL Peer	SSL/TLS peer certificates.
	SSL Revoked	Verbatim revoked SSL/TLS certificates.
Secure Shell (SSHv2)	SSH Host Keys Table	SSHv2 identity private keys. Used for authentication with SSHv2 clients and secure key exchange. A default 1024-bit DSA key is generated automatically if none exists when the device boots. There is no certificate for SSHv2, just private key data.

Behavior of SSH/SSL private keys on Digi devices

Digi devices generate their SSH/SSL self-signed private keys automatically. While this automatic generation is convenient for device users, as they are not required perform any actions regarding the private keys, it presents some security loopholes.

- With self-signed private keys, you must establish trust in a secure environment. That is, if you cannot guarantee that the environment is secure, you must pull the private keys off the Digi device.
- You must know about the certificate before you connect, as opposed to third-party signed certificates, where you only need the third-party certificate.
- The length of Digi's self-signed private keys is **1024** bits. While this length this is adequate for 99.9% of all applications, some people or applications prefer a shorter or longer key.

Using TFTP to load and store certificate information

Using TFTP, you can load and store PEM-formatted certificates into the certificate and private key management tables.

Using HTTP/HTTPS to transfer certificate and key data

On the web, you can use HTTP or HTTPS to transfer certificate and private key data.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-cert=read** to display current certificate management settings, and to **set permissions s-cert=rw** to manage entries in certificate databases. See set permissions for details on setting user permissions for commands.

Syntax

Display certificate management and private key tables

```
certmgmt
```

Set up certificate management and private key tables

```
certmgmt [table={ca|crl|scep_ca|scep_pending|vpn_identity|vpn_key|
   ssl_identity|ssl_key}ssl_peer|ssl_revoked|ssh_key}
  [range={index|index-index|range,range}]
  [action]]
```

***action* is:**

For all tables but SCEP (scep_ca and scep_pending):

```
{display|
remove|
load=ip address:filename|
save=ip address:filename|
password=pem file password|
request={ip address:filename|print}
generate={rsa|dsa:bits {512-4096}}
```

For SCEP tables (scep_ca and scep_pending):

```
action={getca=url
  ca_identifer=ci identifier
accept_ca
range=range
enroll=url
  ca=ca table index
  sig_ca=optional signature ca table index
 enc_ca=optional encryption ca table index
 challenge=challenge password>
  encryption_algorithm={3des|des} (default=3des)
  signature_algorithm={md5|sha1}} (default=md5)
```

Options

table={ca|crl|scep_ca|scep_pending|vpn_identity|vpn_key|ssl_identity|ssl_key}ssl_peer|ssl_revoked|ssh_key}

Identifies a certificate management database table.

ca

Certificate Authority (CA) table.

crl

Certificate A uthority Certificate Revocation Lists (CRL) table.

scep_ca

SCEP CA (Certificate Authority) table.

scep_pending

SCEP Pending Enrollment Requests Table.

vpn_identity

Virtual Private Network (VPN) identity certificates table.

vpn_key

VPN Identity Keys table.

ssl_identity

SSL Identity table.

ssl_key

SSL Identity Keys table.

ssl_peer

SSL Peer table.

ssl_revoked

SSL Revoked table.

ssh_key

SSH Host Keys table.

range={*index*|*index-index*|*range*,*range*}

Identifies a range of entries in a certificate management database table. When **range** is specified as ***index-index***, the table shows the index range to specify for various tables types:

Index range	Applies to table types
1-2	SSL Identity SSL Identity Keys SSH Host Keys
1-4	SCEP Pending Enrollment
1-5	VPN Identity VPN Identity Keys

Index range	Applies to table types
1-8	CA (Certificate Authority) CRL (Certificate Authority) SCEP CA SSL Peer SSL Revoked

action={display|remove|load=*ip address*:*filename*|
save=*ip address*:*filename*|password=*pem file password*|
request={*ip address*:*filename*|print}
generate={rsa|dsa:bits {512-4096}}

For all tables but SCEP (**scep_ca** and **scep_pending**), the action to be performed on the specified database table.

display

Display specific entries or all entries in the specified certificate management database tables.

remove

Removes specific entries or all entries from certificate management database tables.

load=*ip address*:*filename*

Loads certificates via TFTP from the specified server and filename into a certificate management database table

save={*ip address*:*filename*|password=*pem file password*}

Saves certificate management entries via TFTP to the specified file.

request={*ip address*:*filename*|print}

Generates a certificate request and sends it to the TFTP server specified by ***ip address*:*filename*** for the certificate to be signed.

print

Prints out the certificate request so that it can be copied and pasted into an email for emailing the request.

generate={rsa|dsa}:*bits* (512-4096)

Generates a new private key using the specified algorithm (RSA or DSA), or a specified set of bits. This option applies to SSL identity keys, SSH host keys, and VPN identity keys.

action={getca=*url*
ca_identifer=*ci identifier*|
accept_ca
range=*range*
enroll=*url*
ca=*ca table index*
sig_ca=*optional signature ca table index*
enc_ca=*optional encryption ca table index*
challenge=*challenge password*

encryption_algorithm={3des|des}
signature_algorithm={md5|sha1}}

For SCEP tables (**scep_ca** and **scep_pending**), the action to be performed on the specified database table.

getca=*url*

Obtain CA certificates from the SCEP server at the specified URL.

Certificates must be accepted by the operator to be used for any purpose.

ca_identifer=*ca identifier*

Identifies the CA certificate to be obtained from the SCEP server.

accept_ca

Accept the specified CA certificate at the specified URL.This action moves the CA certificate from the SCEP CA to the X.509 CA table.

range=*range*

A range of values in the SCEP table. This option is used to populate empty entries in the SCEP CA table.

enroll=url

Obtain CA certificates from the SCEP server at the specified URL. Additional options for this action include:

ca=*ca table index*

Index number for the CA certificate.

sig_ca=*optional signature ca table index*
enc_ca=*optional encryption ca table index*

There are roles in a certificate enrollment request: The CA that signs the enrollment request, and the CA that encrypts the request. These two options are indices into the CAs in the Digi device's certificate database, and are used to both sign and encrypt the request. This information is typically downloaded from the SCEP CA table. To obtain this information:

1. Enter a **certmgmt** command specifying the **getca** action.

2. Enter another **certmgmt** command, specifying the **accept_ca** action.

sig_ca=*optional signature ca table index*

An optional index number assigned to the CA certificate.

enc_ca=*optional encryption ca table index*

An optional index number associated with the CA that encrypts the request.

challenge=*challenge password*

A simple password that can be used to guard access to certificates.

encryption_algorithm={3des|des}

The encryption algorithm used with the database action.

3des

3DES encryption algorithm, which uses 192-bit keys.

des

DES encryption algorithm, which uses 64-bit keys.

The default is **3des**.

signature_algorithm={md5|sha1}

The authentication algorithm used with the database action.

md5

MD5 authentication algorithm, which uses 128-bit keys.

sha1

SHA1 authentication algorithm, which uses 160-bit keys.

The default is **md5**.

See also

- The X.509 certificate and key management pages in the Digi device web interface at **Management > X.509 Certificate/Key Management**.
- set pmodem: The pmodem feature includes options for specifying authentication, and loading certificates via TFTP and an AT command.

close

Purpose

Closes active connect, Rlogin, and Telnet sessions; that is, sessions opened by **connect**, **rlogin**, or **telnet** commands.

The **close**" command is associated with the sessions displayed by the **status** command.

A **close** command issued without any options closes the current connection.

To issue the **close** command, you must escape the active session. Do this by pressing the escape key defined for your session type. The following table lists default escape keys.

Session Type	Default Escape Keys
Connect	Ctrl+[+Enter
Rlogin	
Telnet	Ctrl+]+Enter

Syntax

```
close [{*|connection number}]
```

Options

*****

Closes all active sessions.

connection number

Identifies the session to close by its session number.

Examples

Close a session identified by number

```
#> close 1
```

Close the current session

```
#> close
```

Close all active sessions

```
#> close *
```

See also

- kill: The **kill** command has a broader effect than **close**, and lets you kill connections from the global list. That is, it is not limited to sessions associated with the current connection.
- status Use this command for information on displaying status information on active sessions.
- connect
- rlogin
- telnet

connect

Purpose

Used to make a connection, or establish a session, with a serial port.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions connect=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

There are several ways to create and manage connections:

Create a single connection

```
connect serial port
```

Create multiple connections

Issue multiple **connect** commands.

Temporarily suspend a connection

Escape the active session by pressing **Ctrl [**.

Temporarily suspend a connection and return to the command line

Press the escape character and then the **Enter** key.

Switch between active sessions (without first escaping to the command line)

Press the escape character and then the number of the session you wish to enter, for example, **Esc+1**.

Pressing the connect escape character twice causes the next session to appear, enabling you to easily page through sessions.

Options

serial port

The number of the port on which to establish a connection.

Example

Create a connection to port 1

```
#> connect 1
```

See also

- close for information on ending a session.
- reconnect for information on reestablishing a port connection.

dhcpserver

Purpose

Used for managing and showing the status of a DHCP server, including managing the leases for IP addresses, restarting, running, and shutting down the DHCP server, and displaying DHCP server status information.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions dhcpserver=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
dhcpserver [deletelease={ip address|all}]
  [restart]
  [run]
  [shutdown]
  [status]
```

Options

deletelease={ip address|all}

Specifies how to handle IP address leases. You can remove leases from the DHCP Server while it is running.

ip address

Removes a specific lease from the DHCP server.

all

Removes all IP address leases from the DHCP server.

Removing a lease causes the associated IP address to be returned immediately to the available address pool. Any IP address in this available address pool can be served in a new lease to a DHCP client.

If you stop or restart the DHCP server, or if you reboot the Digi device, all knowledge of the IP address leases will be lost. All leased addresses, except for reservations, will be returned to the available address pool and may be served in a new lease to a DHCP client.

Static lease reservations always display in the lease list. These reservation leases may be removed, but a new lease will be created immediately. To disable or permanently remove a reservation, use the **set dhcpserver** command.

restart

Restarts the DHCP server.

run

Runs (starts) the DHCP server if it is not already started.

shutdown

Shuts down the DHCP server.

status

Displays DHCP server status information.

Example

Display DHCP server status

```
#> dhcpserver status

Device Networking Status:

  Status for eth0:
    IP address         : 10.30.1.188
    Subnet mask        : 255.255.255.0
    Gateway            : 10.30.1.1
    Using static IP    : yes
  Default gateway      : 10.0.0.1
  Uptime               : 0 days + 21:00:44

DHCP server status: running

  Uptime               : 0 days + 00:00:14

Scopes configured in server:

Scope 1:
  Name                 : eth0
  IP address           : 10.30.1.126
  Subnet mask          : 255.255.255.0
  Starting IP address  : 10.30.1.190
  Ending IP address    : 10.30.1.198
  Routers              : 10.30.1.188
  DNS servers          : 209.183.48.10, 209.183.48.11
  Lease duration       : 3600 (seconds)
  Offer delay          : 500 (milliseconds)
  Addr conflict detect : disabled
  Send DNS proxy       : enabled
Address reservations:
    Reservation 1:
      IP address           : 10.30.1.135
      Client ID            : 00:40:9D:24:73:F8
      Lease duration       : 3600 (seconds)
    Reservation 2:
      IP address           : 10.30.1.192
      Client ID            : 02:40:9D:24:73:F8
      Lease duration       : using scope lease duration
    Reservation 3:
      IP address           : 10.30.1.195
```

```
    Client ID           : 00:09:26:19:51:05
    Lease duration      : using scope lease duration
  Reservation 4:
    IP address          : 10.30.1.196
    Client ID           : 00:09:26:19:51:06
    Lease duration      : using scope lease duration
  Reservation 5:
    IP address          : 10.30.1.197
    Client ID           : 00:09:26:19:51:07
    Lease duration      : using scope lease duration
Address exclusions:
  none configured
Lease Records:
                  Client ID         Lease Time in Seconds
  IP Address      (MAC Address)     Duration   Remaining  Lease Record Status
  --------------- ----------------- ---------- ---------- ------------------
  10.30.1.135     00:40:9D:24:73:F8 3600       1834       Reserved (active)
  10.30.1.192     02:40:9D:24:73:F8 3600       N/A        Reserved (inactive)
  10.30.1.195     00:09:26:19:51:05 3600       N/A        Reserved (inactive)
  10.30.1.196     00:09:26:19:51:06 3600       N/A        Reserved (inactive)
  10.30.1.197     00:09:26:19:51:07 3600       N/A        Reserved (inactive)
```

Delete a lease

```
dhcpserver deletelease=10.30.1.135
```

Delete all leases

```
dhcpserver deletelease=all
```

Output

Lease status values

Following are descriptions of the lease status values. The amount of time that a lease table entry will remain in each state also is stated. Note that after a lease is deleted, the associated IP address is returned to the available address pool.

Assigned (active)

A lease is currently assigned and active for the given client. The client may renew the lease, in which case the lease remains in this state.

Assigned (expired)

A lease has expired and is no longer active for the given client. A lease in this state will remain for 4 hours, after which it is deleted. If the same client requests an IP address before the lease is deleted, it will be given the same IP address previously served to it.

Reserved (active)

A lease for an address reservation is currently active for the given client. A reservation lease will remain indefinitely, although the status may alternate between active and inactive.

Reserved (inactive)

A lease for an address reservation is currently inactive for the given client. A reservation lease will remain indefinitely, although the status may alternate between active and inactive.

Reserved (unavail)

A lease for an address reservation was offered to a client, but that client actively declined to use the IP address. Typically this is because the client determined that another host on the same subnetwork is already using that IP address. Upon receiving the client's decline message, the DHCP server will mark the address as unavailable. The lease will remain in this state for 4 hours, after which it is reverts to the **Reserved** (inactive) status.

Offered (pre-lease)

A lease has been offered to the given client, but that client has not yet requested that the lease be acknowledged. It may be that the client also received an offer from another DHCP server, in which case this offer will expire in approximately 2 minutes. If the client requests this lease before that 2 minute interval elapses, this lease will change status to **Assigned**.

Released

A lease was previously assigned to the given client, but that client has proactively released it. A lease in this state will remain for 1 hour, after which it is deleted. If the same client requests an IP address before the lease is deleted, it will be given the same IP address previously served to it.

Unavailable Address

A lease was offered to a client, but that client actively declined to use the IP address. Typically this is because the client determined that another host on the same subnetwork is already using that IP address. Upon receiving the client's decline message, the DHCP server will mark the address as unavailable. The lease will remain in this state for four hours, after which it is deleted.

This status may also occur if the DHCP Server determines that the IP address is in use before it offers the address to a client. See the **set dhcpserver** command option **conflictdetect**.

See also

- set dhcpserver
- set dnsproxy
- The web interface’s help text for Network Settings, which includes information on configuring DHCP server settings and managing DHCP servers.

display accesscontrol

Purpose

Displays access control status information.

Required permissions

For Digi products with two or more users, you must set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display accesscontrol
```

Examples

```
#> display accesscontrol

Access Control Status:

 ACL (IP source filtering) feature:   disabled
 Automatically allow local subnets:   disabled

 Dynamic access control lists:
   Number of dynamic ACL addresses:   4
    Entry  Address          Entry  Address
       1:  10.8.115.10         3:  10.10.8.64
       2:  127.0.0.1           4:  10.10.8.62

   Number of dynamic ACL subnets:     2
    Entry  Network          Mask
       1:  10.8.0.0         255.255.0.0
       2:  127.0.0.1        255.255.255.255
 Static access control lists:

   Number of static ACL addresses:    0
   Number of static ACL subnets:      0

 Local protocol/port exceptions:
   Number of local exceptions:        2
    Entry  Protocol   Port   Description
       1:    UDP      2362   ADDP Service
       2:    UDP        68   [BOOTP/DHCP client]

 ACL Statistics:
   IP datagrams examined:             0
   IP datagrams accepted:             0
   IP datagrams discarded:            0
```

display arp

Purpose

Displays ARP table entries.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display arp
```

Examples

```
#> display arp

 ARP Table:

 IPAddress       HwAddress         HWtype  HWL  Refs Iface    State TTL
 10.8.1.1        00:00:0C:07:AC:01 ether   6    2    eth0     IPV4  420000
 10.8.1.2        00:12:00:1D:31:FF ether   6    1    eth0     IPV4  900000
 10.8.1.3        00:12:00:1D:32:BF ether   6    1    eth0     IPV4  900000
 10.8.8.52       08:00:20:80:17:0D ether   6    1    eth0     IPV4  240000
 10.8.8.54       00:50:8B:DD:18:EB ether   6    1    eth0     IPV4  720000
 10.8.8.56       00:1A:64:21:04:46 ether   6    1    eth0     IPV4  840000
 10.8.8.57       00:0C:29:E7:EE:C1 ether   6    1    eth0     IPV4  720000
 10.8.8.58       00:C0:9F:25:04:52 ether   6    1    eth0     IPV4  480000
 10.8.8.70       00:07:E9:47:82:87 ether   6    1    eth0     IPV4  660000
 10.8.8.72       00:07:E9:47:91:7E ether   6    1    eth0     IPV4  900000
 10.8.8.76       00:11:11:E2:03:A5 ether   6    1    eth0     IPV4  780000
 10.8.8.78       00:07:E9:4D:3D:FE ether   6    1    eth0     IPV4  720000
 10.8.8.82       00:90:27:5A:A7:5D ether   6    1    eth0     IPV4  660000
 10.8.8.86       00:50:DA:1B:9A:ED ether   6    1    eth0     IPV4  360000
 10.8.8.88       00:12:3F:72:E3:E2 ether   6    1    eth0     IPV4  420000
 10.8.8.90       00:12:3F:72:E3:98 ether   6    1    eth0     IPV4  600000
 10.8.9.1        00:02:55:4F:69:D9 ether   6    1    eth0     IPV4  180000
 10.8.9.45       00:30:6E:4B:19:FD ether   6    1    eth0     IPV4  120000
 10.8.9.101      00:00:C0:46:6C:93 ether   6    1    eth0     IPV4  600000
 10.8.16.5       00:0F:FE:80:6D:BA ether   6    1    eth0     IPV4  660000
 10.8.16.11      00:0F:FE:92:9C:77 ether   6    1    eth0     IPV4  900000
 10.8.16.13      00:60:0C:01:AE:9C ether   6    1    eth0     IPV4  600000
```

See also

- set network
- show

display buffers

Purpose

Displays the contents of a port buffer, or transfers the contents of a port buffer to a server running Trivial File Transfer Protocol (TFTP). Enable port buffering using the **set buffer** command (see set buffer). Contents display in log form.

Required permissions

For Digi products with two or more users, permissions must be set to one of the following:

- For a user to display the contents of a port buffer for the line on which they are logged in: **set permissions buffers=r-self** or higher.
- For a user to display the contents of a port buffer for any line: **set permissions buffers=read** or higher.

See set permissions for details on setting user permissions for commands.

Syntax

```
display buffers [port=range] {[screen] [lines=number]
   [tail=number]   | tftp=server:filename}
```

Options

port=*range*

The port or ports to which the command applies. Optional on a single-port device.

screen

Displays the port buffer contents on the screen when screen is specified.

lines=*number*

The number of lines of data to display at a time when the **screen** option is specified. Use 0 to indicate continuous flow.

tail=*number*

The total number of lines in the buffer to be displayed. The number is calculated from the end of the buffer counting back.

tftp=server:*filename*

server

The IP address or DNS name of a server running TFTP to which buffer information should be transferred.

filename

The name to use for the file that transferrs to the TFTP server. If the **port** option specifies more than one port, one file transferrs for each port. The filename for each port will be ***filename_n***, where ***n*** is the port number.

Examples

Display port buffering information on the screen

```
#> display buffers port=2 screen lines=32 tail=30
```

Output buffering information to a TFTP server

```
#> display buffers port=2 tftp=192.168.1.1:port_ouput
```

Output multi-\port buffering information to a TFTP server

```
#> displady buffers port=2-3 tftp=192.168.1.1:port_ouput
```

Note Port 2 buffering information goes to file port_output_2 and port 3 buffering information goes to file port_output_3.

See also

set buffer

display carriers

Purpose

Displays a list of the GSM carriers that are broadcasting in the area of the Digi device and available for use as a carrier; that is, carriers that have towers and equipment in the area. GSM carriers broadcast that they are available, and the phone/modem selects the best one.

The output from this command includes the carrier names and associated numbers that represent the carrier (the mobile country code (MCC), concatenated with the mobile network code (MNC), in this manner: **<MCC>(+)<MNC>**. The numbers can be then used for the **set mobile carrier=*mobile country code+mobile network code*** command.

Before using this command, you must disable the cellular module.

1. Issue the command **set mobileppp state=disable**.
2. Kill the PPP session by issuing the **who** command. In the ID field of the command output, locate the index number for the PPP session is displayed.
3. Issue the **kill** command, specifying the index number for the PPP session, for example, **kill 3**
4. Issue the **display carriers** command.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. In addition, the commands noted in the procedure above, **set mobilepp**, and the **who** command, have their own permissions and are noted in their respective command descriptions. See set permissions for details on setting user permissions for commands.

Syntax

```
display carriers
```

Example

```
#> set mobileppp state=disable
#>
#> who

ID  From                  To                    Protocol              Sessions
--  ------------------    ------------------    --------------------  --------
1                                               ppp [mobile init]
2   local                 USB                   NMEA GPS Processor
3   10.8.16.105           local shell           telnet

#>
#> kill 1

Connection 1 : Killing connection...

#>
```

```
#> display carriers

Active Carrier List

Carrier Name             Network ID              Status          Service
--------------------     -----------             ----------      --------
AT&T                     310410                  Available       2G
AT&T                     310410                  Current         3G
T-Mobile                 31026                   Available       2G
```

See also

- set mobile
- set mobileppp
- who
- kill

display ddns

Purpose

Displays status information for the Dynamic DNS (DDNS) service.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display ddns
```

Examples

```
#> display ddns

 Current IP address: 166.131.0.158 (mobile0)

 No previous DDNS service update status is available.

 Most recent DDNS service update log message:
    IP address for “mobile0” is now 166.131.0.158, but no DDNS update is needed
(updates are disabled).

 To view current DDNS service settings, use “show ddns”.
```

See also

set ddns for information on the DDNS service.

display device

Purpose

Displays general product information, including the product name, MAC address, boot, post, and firmware versions, memory usage, CPU utilization, and uptime, or the amount of time since the device was last booted. The information displayed by this option is the same as that displayed by the **info device** command (see info device).

100% CPU Utilization may indicate encryption key generation is in-progress

There may be instances when a **display device** command returns a CPU utilization of 100%. A CPU usage this high may indicate that encryption key generation is in-progress.

On initial boot, the Digi device generates some encryption key material:

- RSA key for SSL/TLS operations
- DSA key for SSH operations.

This key-generation process can take as long as 40 minutes to complete. Until the corresponding key is generated, the Digi device will be unable to initiate or accept that type of encrypted connection. It will also report itself as 100% busy but, since key generation takes place at a low priority, the device still functions normally. On subsequent reboots, the Digi device uses its existing keys and will not need to generate another unless a reset to factory defaults is done, which causes a new key to be generated on the next reboot.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display device
```

Options

None.

Examples

```
#> display device

Device Information:

Product              : ConnectPort X4
MAC Address          : 00:40:9D:50:EA:E5
Firmware Version     : 2.14.1.5 (Version 82001536_K_SA5 09/30/2011)
Boot Version         : 1.1.3 (release_82001975_D)
Post Version         : 1.1.3 (release_82001753_G)
Product VPD Version  : release_82002003_B
Product ID           : 0x0081
```

```
Hardware Strapping    : 0x0044
CPU Utilization       : 8 %
Uptime                : 17 days, 1 hour, 25 minutes, 28 seconds
Current Date/Time     : Fri Mar 16 05:56:44 2007
Total Memory          : 33554432
Used Memory           : 13515372
Free Memory           : 20039060
Total Flash FileSys   : 3999744
Used Flash FileSys    : 363520
Free Flash FileSys    : 3636224
```

See also

info device for descriptions of the information displayed for this command.

display dnsserver

Purpose

Displays the current list of DNS servers configured in the Digi device's network stack for name resolution.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display dnsserver
```

Examples

```
#> display dnsserver

 DNS Server List:

 Index    IP Address
   0      10.10.8.64
   1      10.10.8.62
   2      0.0.0.0
   3      0.0.0.0
```

See also

- set network to set the DNS priority list (**set network dnspriority=...**) and set static DNS server IP addresses.
- show: The **show network** command displays the DNS priority list and DNS server IP addresses (**set network dns1=*ipaddr1* dns2=*ipaddr2***).
- display pppstats to display **mobile0** (cellular) DNS IP addresses.
- display wimax to display WiMAX service DNS IP addresses.

display failover

Purpose

Displays IP network failover status and statistics.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display failover
```

Examples

In the following example, the Ethernet interface, eth0, has been configured for IP failover.

```
#>display failover
Failover is on/enabled, which overrides the non-failover gateway configuration.
Current Default Gateway Status:
  Gateway interface : eth0
  Gateway address   : 192.168.250.1
  Configured by     : failover
Current Network Failover Status:
  Failover state           : on
  Fallback to non-failover : on
  Priority  Interface       Status              Gateway         State  Tests
     1      mobile0         2 (up)              68.28.89.85      on      0
     2      eth0            1 (responding)      192.168.250.1    on      1
     3      wln0            2 (up)              192.168.33.1     on      0
Current Network Gateway Status (Non-Failover):
  Priority  Interface       Status      Gateway
     1      mobile0         1 (up)      68.28.89.85
     2      eth0            1 (up)      192.168.250.1
     3      wln0            1 (up)      192.168.33.1
Current Failover Link Test Statistics:
                    Test         Test       Bypass  Consecutive     Link Not
  Interface       Success      Failure        Test     Failures   Responding
  mobile0               0            0           0            0            0
  eth0                  1            0           0            0            0
  wln0                  0            0           0            0            0
```

These examples show using the **display failover** command to display the network interface list when the **mobile0** (cellular) interface is up and down.

When **mobile0** interface is up:

```
#> display failover

Failover is off/disabled, which enables the non-failover gateway configuration.
```

```
Current Default Gateway Status:

  Gateway interface : mobile0
  Gateway address   : 10.0.0.1
  Configured by     : non-failover

Current Network Failover Status:

  Failover state           : off
  Fallback to non-failover : on

  Priority  Interface       Status              Gateway         State  Tests
     1      mobile0         2 (up)              10.0.0.1         on      0
     2      eth0            2 (up)              10.30.1.1        on      0

Current Network Gateway Status (Non-Failover):

  Priority  Interface       Status      Gateway
     1      mobile0         1 (up)      10.0.0.1
     2      eth0            1 (up)      10.30.1.1

Current Failover Link Test Statistics:

                     Test         Test       Bypass  Consecutive     Link Not
  Interface       Success      Failure         Test     Failures   Responding
  mobile0               0            0            0            0            0
  eth0                  0            0            0            0            0
```

When **mobile0** interface is down:

```
#> display failover

Failover is off/disabled, which enables the non-failover gateway configuration.

Current Default Gateway Status:

  Gateway interface : eth0
  Gateway address   : 10.30.1.1
  Configured by     : non-failover

Current Network Failover Status:

  Failover state           : off
  Fallback to non-failover : on

  Priority  Interface       Status              Gateway         State  Tests
     1      mobile0         4 (down)            0.0.0.0          on      0
     2      eth0            2 (up)              10.30.1.1        on      0

Current Network Gateway Status (Non-Failover):

  Priority  Interface       Status      Gateway
     1      mobile0         0 (down)    0.0.0.0
     2      eth0            1 (up)      10.30.1.1

Current Failover Link Test Statistics:

                     Test         Test       Bypass  Consecutive     Link Not
```

Interface	Success	Failure	Test	Failures	Responding
mobile0	0	0	0	0	0
eth0	0	0	0	0	0

See also

set failover for more information on the IP network failover feature.

display gpio

Purpose

Displays General Purpose I/O (GPIO) signals.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display gpio
```

Examples

```
#> display gpio

GPIO States:

  pin# state
     1    on
     2    on
     3    on
     4    on
     5    on
```

See also

set gpio

display gps

Purpose

Shows the current position of the GPS unit connected to the Digi device, and information about the readings taken by the GPS unit.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display gps
```

Examples

```
#> display gps

  Statically Configured Parameters:

    Latitude                  : 40.730000
    Longitude                 : -91.961998
```

See also

- set geofence
- set position

display dcloud

Purpose

Displays the status of the connection to Device Cloud.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display dcloud
```

Examples

Display Device Cloud connection status for a ConnectPort X4 device:

```
#> display dcloud

    Device Cloud Connection Status:

    Device Type          : ConnectPort X4
    Device ID            : 00000000-00000000-00409DFF-FF36DEA8
    Status               : Connected to 50.56.41.153:3199 from
166.130.115.207:34186
    Server URL           : en://login.etherios.com
    Connection Type      : Device-initiated
    Connection Method    : TCP/SSL
    Connected For        : 3 days + 00:29:58
    Receive Idle For     : 00:14:52
    Send Idle For        : 00:14:53
```

Display Deviced Cloud connection status for a ConnectPort X5 Iridium device

Note the additional statistics displayed for the Iridium® satellite modem.

```
#> display dcloud

    Device Cloud Connection Status:

    Device Type          : ConnectPort X5 R
    Device ID            : 00000000-00000000-004F3FF-FF03A8C0
    Status               : Connected to 50.56.41.156:3199 from 10.52.18.91:43378
    Server URL           : en://login.etherios.com
    Connection Type      : Device-initiated
    Connection Method    : TCP/SSL
    Connected For        : 00:00:09
```

```
Receive Idle For     : 00:00:09
Send Idle For        : 00:00:09

Device Cloud Iridium Status:

Bytes Received                : 0
Bytes Sent                    : 0
Messages Received             : 0
Messages Sent                 : 0
Successful Sends              : 0
Failed Sends                  : 0
Queued Send Messages          : 0
Queued Receive Messages       : 0
Failed Polls                  : 0
Successful Polls              : 0
```

See also

- set devicesecurity
- set mgmtconnection
- set mgmtglobal
- set mgmtnetwork

display ikesa

Purpose

Displays the IKE Security Association (SA) table for Virtual Private Network (VPN) tunnels.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display ikesa
```

Examples

```
#> disp ikesa

IKE SA Table:

Source IP       Dest. IP        Enc    Hash  Auth    DH  Life(s)   Life(KB)
66.165.177.14   66.165.177.14             nul 0         0
```

See also

- display ikespd
- display ipsecspd
- display sadb
- display spd
- display vpn
- set vpn
- vpn

display ikespd

Purpose

Displays the Internet Key Exchange (IKE) Security Policy Database (SPD) entries for VPN tunnels.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display ikespd
```

Examples

```
#> disp ikespd

IKE SPD Table:

Source IP       Dest. IP        Hash  Enc   Auth    DH  Life(s)   Life(KB)
166.203.137.94:066.165.177.14:0 md5   aes   psk     2   240       32768
```

See also

- display ikesa
- display ipsecspd
- display sadb
- display spd
- display vpn
- set vpn
- vpn

display ipsecspd

Purpose

Displays the IPSEC Security Policy Database (SPD) entries defined for VPN tunnels.

Note This is the information that was formerly displayed by the **display spd** command.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display ipsecspd
```

Examples

```
#> disp ipsecspd

IPSEC SPD Table:

Idx, Selector(local ip:port,remote ip:port, protocol),Inner policy, Outer policy,
    Protect Mode  Hash  Enc   Protect Mode  Hash
  0 any, any, any                    bypass
  1 10.13.11.96:0, 0.0.0.0:0, any    esp     tunnl sha1  aes
  2 10.13.11.96:0, 0.0.0.0:0, any    esp     tunnl sha1  aes
  3 any, 127.0.0.1:0, any            bypass
  4 any, 10.13.11.96:0, any          bypass
  5 any, 166.203.137.88:0, any       bypass
```

See also

- display ikesa
- display ipsecspd
- display sadb
- display spd
- display vpn
- set vpn
- vpn

display iridium

Purpose

Displays current state information in the Iridium® satellite modem subsystem, including such elements as:

- Power state
- Modem serial number
- Modem software revision
- Signal strength
- Network availability

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display iridium
```

Examples

```
#> disp iridium

Iridium Satellite Modem State:

  Power                : on

  Serial number        : 300234010129780
  Manufacturer         : Iridium
  Modem Model          : IRIDIUM 9600 Family SBD Transceiver
  Software version     : TA10003
  Signal strength      : 0
  Network available    : no
```

See also

- info iridium
- iridium
- set trace: The **iridium** option captures debugging information and error conditions from the Iridium satellite modem subsystem.

display logging

Purpose

Displays contents of the event log. This log records events throughout the Digi device's system, such as:

- Starting or resetting the Digi device
- Configuring features
- Critical actions performed by various interfaces and subsystems
- Starting applications and other important events.

Many but not all Digi devices have the event log feature. The event log is always enabled and is not user-configurable. When the Digi device operates in an unexpected manner, you can send the log entries to Digi for analysis by Technical Support and Engineers. The event log cannot be turned off, so that Digi receives an accurate view of all aspects of the operation of the device. Command options control how much or which portion of the event log is displayed, and clear the event log.

The event log is a 64KB circular buffer that can contain a varying number of log entries. A log entry can have a maximum number of bytes per entry. Based upon the size of the entries, the number of entries kept in the log varies. When the log "overflows," the oldest entries are overwritten with new ones, so the history is incomplete.

The event log is maintained in RAM, and there is no history across reboots of the device. Logging is always on, and there is no way for a user to enable/disable, adjust what is logged, or adjust the level of detail.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display logging [action={clear|display}]
  [head=lines]
  [tail=lines]

display logging [options...]
```

Where:

```
options:
    [action={clear|display}]
```

Options to display the full event log (action=display):

```
    [timeformat={auto|datetime|raw|uptime}
```

Options to display a portion of the event log:

```
[head=lines]
[tail=lines]
```

Options

action={clear|display}

Perform an action on the event log.

clear

Clears the event log.

display

Displays the full event log

The default is **display**.

timeformat={auto|datetime|raw|uptime}

The format for displaying time in event log messages.

auto

Automatically select the time format (default).

datetime

Display time in a date/time format of ***YYYY-MM-DD hh:mm***:. For example

```
2008-08-23 09:26:51
```

raw

Display time as a raw value: show the time as an integer number of seconds.

uptime

Display time in an uptime format: ***+N*** days ***hh:mm:ss***. For example:

```
+2 days 13:06:32
```

head=lines

Displays the first lines of the event log, that is, the oldest lines. The ***lines*** value must be 1 or more.

tail=lines

Displays the last lines of the event log; that is, the most recent lines. The ***lines*** value must be 1 or more.

The ***lines*** value can be followed by a **+** character to request that the last ***lines*** event log entries be displayed, after which the command waits for and display new event log messages as they are added to the log, in other words, continue tailing the log. Enter **Ctrl+C** to end the command.

For example, appending a **+** to the **tail** option value displays the last 10 lines of the log, after which the Digi device continues waiting for and displaying, new log entries as they arrive.

```
#> display logging tail=10+
>>> Tailing the event log (Ctrl-C to cancel)...
2011-10-24 18:19:40 mobile     Mobile CID changed to 14349 from 14343.  MCC: 310
MNC: 410
2011-10-24 18:19:42 mobile     RSSI: -102 dBm
2011-10-24 18:19:46 mobile     RSSI: -98 dBm
2011-10-24 18:20:04 mobile     RSSI: -94 dBm
2011-10-24 18:20:25 mobile     RSSI: -83 dBm
2011-10-24 18:20:28 mobile     RSSI: -98 dBm
2011-10-24 18:20:31 mobile     Mobile CID changed to 14343 from 14349.  MCC: 310
MNC: 410
2011-10-24 18:20:31 mobile     RSSI: -84 dBm
2011-10-24 18:20:57 mobile     Mobile CID changed to 14349 from 14343.  MCC: 310
MNC: 410
2011-10-24 18:20:57 mobile     RSSI: -99 dBm
>>> Tail canceled.
2011-10-24 18:21:15            [End of Event Log at +3 days 02:27:07]
```

Entering CTRL+C to cancel the continuous tail. The last two lines are output. This is similar to ending a **ping**.

Examples

Display the event log

```
#> display logging

2011-10-04 18:06:41 boot       Unit started.
2011-10-04 18:06:41 time       RTC initialized: Tue Oct  4 18:06:41 2011.
2011-10-04 18:06:41 time       Applying system time adjustment threshold of 60
seconds.
2011-10-04 18:06:41 time       Lost time source recovery is enabled.
2011-10-04 18:06:41 system     Product: ConnectPort X4 (ID 0x0085).
2011-10-04 18:06:41 system     Firmware Version: 2.14.1 (Version Skipjack 07/14/
2011 02:43:01 CDT).
2011-10-04 18:06:41 system     Boot Version: 1.1.3 (release_82001975_C).
2011-10-04 18:06:41 system     Post Version: 1.1.3 (release_82001753_D).
2011-10-04 18:06:41 system     Product VPD Version: release_82002010_B.
2011-10-04 18:06:41 system     Hardware Strapping: 0x0044.
2011-10-04 18:06:41 failover   Failover feature is started.
2011-10-04 18:06:41 failover   Failover state is OFF, fallback is ON.
2011-10-04 18:06:41 dnsproxy   DNS Proxy start-up succeeded.
2011-10-04 18:06:41 failover   Link testing for "eth0" is ON.
2011-10-04 18:06:41 system     Network Port Scan Cloak feature is enabled.
2011-10-04 18:06:41 dcloud     Device ID: 00000000-00000000-00409DFF-FF492A01
2011-10-04 18:06:41 dcloud     Vendor ID: 0xFE000000
2011-10-04 18:06:41 dcloud     Device Type: ConnectPort X4
2011-10-04 18:06:42 time       Jump to new baseline at uptime 1 (src 5, rank 50,
delta 1317751601).
2011-10-04 18:06:42 mesh       Radio version: HW 0x1a41 FW 0x3128
2011-10-04 18:06:47 failover   Interface "eth0" status is now 4 (down).
2011-10-04 18:06:47 failover   Interface "eth0" status is now 2 (up).
2011-10-04 18:06:49 dcloud     Starting device-initiated connection.
```

```
2011-10-04 18:06:49 dcloud     Finding Service, URL=en://login.etherios.com
2011-10-04 18:06:49 dcloud     Trying TCP.
2011-10-04 18:06:49 dcloud     Connected at socket level: Local=10.8.16.111:4813
3 Remote=50.56.41.153:3197.
2011-10-04 18:06:49 dcloud     Keep-alive parameters are: RX=60 TX=60 wait=3.
2011-10-04 18:06:49 dcloud     Connected Using TCP.
2011-10-05 15:50:46 dcloud     Error 116 on TCP socket read.
2011-10-05 15:50:46 dcloud     Disconnected from server.
2011-10-05 15:50:46 dcloud     Device-initiated connection failed, return=-6
2011-10-05 15:50:46 dcloud     Connection Error: Connection to Server Closed.
2011-10-05 15:50:46 dcloud     Will reconnect to server in 10 seconds.
2011-10-05 15:50:56 dcloud     Starting device-initiated connection.
2011-10-05 15:50:56 dcloud     Finding Service, URL=en://login.etherios.com
2011-10-05 15:50:56 dcloud     Trying TCP.
2011-10-05 15:51:05 dcloud     Connected at socket level: Local=10.8.16.111:3809
3 Remote=50.56.41.153:3197.
2011-10-05 15:51:05 dcloud     Keep-alive parameters are: RX=60 TX=60 wait=3.
2011-10-05 15:52:48 dcloud     Connected Using TCP.
2011-10-21 15:17:42 time       Jump to new baseline at uptime 1458721 (src 5,
rank 50,
delta -60).
2011-10-21 23:11:32            [End of Event Log at +17 days 05:05:51]
```

Display the last ten lines from the event log

```
#> display logging tail=10

1970-01-01 00:00:08 dhcpserver No scopes/interfaces started successfully.
1970-01-01 00:00:08 dhcpserver Stopping DHCP Server.
1970-01-01 00:00:08 dhcpserver DHCP Server stopped.
1970-01-01 00:00:10 mobile     Module reset (type 25: Sierra Wireless MC87X5).
1970-01-01 00:00:10 mesh       Radio version: HW 0x1941 FW 0x2141
1970-01-01 00:00:24 mobile     Radio temperature: 33 (C)
1970-01-01 00:00:24 mobile     Mobile LAC changed to 0 from -1.  Country: 0,
Network:
1970-01-01 00:00:24 mobile     Mobile CID changed to 0 from -1.  Country: 0,
Network:
1970-01-01 00:00:24 mobile     SIM Status changed to SIM removed (2) from SIM OK
(0)
1970-01-01 00:01:02 mobile     RSSI: -110 dBm
1970-01-02 06:44:14            [End of Event Log]
```

Clear the event log

```
#> display logging action=clear
```

See also

- You can also display the event log in the web interface by selecting **Management > Logging**.
- display techsupport
- The “info” commands. These commands display various device statistics that may aid in troubleshooting your Digi product.
- set trace

- The Troubleshooting chapter of the *Digi Connect Family and ConnectPort TS Family User Guide* for your Digi product.
- The Digi Support web page to contact Technical Support, search the Digi knowledge base, ask a question on the Support forum, and get diagnostics and utilities.

display memory

Purpose

Displays memory usage information, including general memory, network memory, and streams memory usage.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display memory
```

Examples

```
#> display memory

Memory Usage:

 Main Memory :
    Total Memory         : 16777216
    Used Memory          : 11598844
    Free Memory          : 5178372

 Network Memory :          Current     Maximum
    Allocated            : 200556      208604
    In use               : 176840      192868
```

See also

info device

display mobile

Purpose

Displays mobile (cellular modem) status information, including network registration status, signal strength, subscriber and equipment information, and SIM card information. This command applies to cellular-enabled Digi products only.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display mobile
```

Examples

Display mobile/cellular device status and ID information

```
#> display mobile

Mobile Status:

 Network Registration Information:
     Registration Status    : 1, Registered (Home Network)
     Location Area Code     : 0x55E6 (21990)
     Cell ID                : 0x03F2AD37 (66235703)
     Signal Strength        : 4 of 5 bars (-78 dBm)

 Subscriber and Equipment Information:
     Mobile Version         : 1.1
     IMSI                   : 310410158909813
     ICCID                  : 89014102211589098131
     Phone Number           : N/A
     Manufacturer ID        : Qualcomm Incorporated
     Model ID               : Qualcomm Gobi3000
     Serial Number          : 001084000012420
     Revision ID            : D3200-STSUGN-1575  1  [Nov 22 2010 09:00:00]
     Modem MEID             : A10000176416BF
     Mobile Country Code    : 310
     Mobile Network Code    : 410
     Roaming Status         : Home Network
     Band Class             : WCDMA 850
     Channel                : 4359
     Data Attach Status     : Attached
     Transmit Rate          : 34400 bps
     Receive Rate           : 25600 bps
     Cell Type              : HSDPA, HSUPA, WCDMA
```

```
     Network Name           : AT&T
     SIM PIN Status         : Ready
     User Band Selection    : Automatic
     User Carrier Selection : Automatic
     Signal Quality         : -7.0 dBm

 SIM Information:
     Slot  IMSI and ICCID        Phone Number    Active  Status          PIN
Status
     ----  --------------------  --------------  ------  --------------  --------
--
     1     310410158909813       N/A             yes     primary         Ready
           89014102211589098131
     2     IMSI: N/A             N/A             no      secondary       N/A
           ICCID: N/A
```

See also

- display carriers
- display pppstats to display statistics associated with the SureLink feature.

display nat

Purpose

Displays Network Address Table (NAT) status information.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display nat
```

Examples

```
#> display nat

IP forwarding is enabled.

NAT Status (mobile0):

  NAT is running for this interface.

  NAT General Status:
    Current number of NAT rules: 1 (maximum 1)
    Configured limit for number of NAT rules: 256
    GRE translation: enabled
    ESP translation: disabled
    Allow TCP port forwarding rule conflicts: no
    Allow UDP port forwarding rule conflicts: no
    Allow FTP port forwarding rule conflicts: no
    NAT currently suspended: no
    NAT rule allocation counters:
      Successful:         1
      Failed (limit):     0
      Failed (no memory): 0
  NAT Rules for Address and Port Translation:
    NAPT *=166.130.0.159 : 51247 of 49152-53247

  Port/Protocol Forwarding Rules:
    None.

  Port Forwarding Exclusion Rules:
    None.

  Current NAT Conversations (Dynamic Rules):
    None.

NAT Statistics:
```

```
          Translations of Addresses and Ports:    Conversations Created:
Proto     Private to Public  Public to Private    Since Boot   Current
-----     -----------------  -----------------    ----------  ----------
 TCP                  0                  0              0         0
 UDP                  0                  0              0         0
 ICMP                 0                  0              0         0
 GRE                  0                  0              0         0
 ESP                  0                  0              0         0
 all                  0                  0              0         0
```

See also

set nat

display netdevice

Purpose

Displays the active network device interfaces on the system, for example, PPP and Ethernet interfaces, and their status, such as **Closed** or **Connected**.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

display netdevice

Examples

```
#> display netdevice
 Device Table:

 Name     PhysAddr          Status
 vpn4     NA                closed
 vpn3     NA                closed
 vpn2     NA                closed
 vpn1     NA                closed
 vpn0     NA                closed
 eth0     00:40:9D:29:8D:07 connected
 vrrp7    NA                closed
 vrrp6    NA                closed
 vrrp5    NA                closed
 vrrp4    NA                closed
 vrrp3    NA                closed
 vrrp2    NA                closed
 vrrp1    NA                closed
 vrrp0    NA                closed
 LOOPBACK NA                connected


 Device Entry IP Configuration:
 Name     Family mHome Type   Status      IPAddress
 eth0     IPv4   0     manual configured  10.8.115.10/16
 LOOPBACK IPv4   0     manual configured  127.0.0.1/32
```

See also

- The **set** commands for configuring network device interfaces.
- The **display** commands for displaying current network device interface states.

- The Network Port Scan Cloaking feature allows you to configure a Digi device to ignore (discard) received packets for services that are hidden or not enabled and network ports that are not open. Scan cloaking can be performed on all connection requests, or for certain types of requests for particular network interfaces. See set scancloak.

display orbcomm

Purpose

Displays information about the state of the ORBCOMM® satellite modem.

The **state** setting is always displayed. When the **state** setting is disabled, the only information available is whether the device is powered on or not. When the **state** setting is enabled, the system attempts to query for several types of information, including the software version of the modem and network readiness.

Note on network readiness

If an ORBCOMM satellite modem attempts to transmit on the ORBCOMM network in a way that is inappropriate, typically because it is not yet provisioned on that network, the network can shut down the modem remotely – and the network readiness value will be **FORCED SILENT**. When a modem is in this state, the state must be cleared before any transmission will occur, using the **orbcomm action=factory** command. For more information, see orbcomm.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display orbcomm
```

Examples

```
#> display orbcomm

Operating System Management of ORBCOMM Satellite Modem: disabled

ORBCOMM Satellite Modem State:

  Power                : off
  Power Cycle Count    : 0

#> disp orb

Operating System Management of ORBCOMM Satellite Modem: enabled

ORBCOMM Satellite Modem State:


  Power                : off
  Power Cycle Count    : 0


#> disp orb
```

```
Operating System Management of ORBCOMM Satellite Modem: enabled

ORBCOMM Satellite Modem State:

  Power                : on
  Power Cycle Count    : 1

  Serial number        : M05000000363
  Firmware revision    : 2.0.0.6
  Hardware revision    : 1.04

  Signal strength      : 0 dBm
  Signal/Noise         : 0 dB
  Doppler shift        : 0 Hz
  Satellite in view    : 0
  Downlink channel     : 0
  Temperature          : 31 C
  Network readiness    : ready
```

See also

- info orbcomm
- orbcomm
- set orbcomm
- set trace: The **orbcomm** option captures debugging information and error conditions from the ORBCOMM satellite modem subsystem.

display passthrough

Purpose

Displays status of the IP pass-through mode, which is enabled by the set passthrough command.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display passthrough
```

Examples

The following is an example command when IP pass-through is not enabled (that is, the Digi device is operating in its normal mode):

```
#> display passthrough

IP Pass-through Mode Status:

    boot state                 : off
    current status             : inactive
    current ip address         : 0.0.0.0
    current gateway            : 0.0.0.0
    current subnet mask        : 0.0.0.0
    current proxy arp          : inactive
Add an example w/ IP passthrough mode already set up.
```

See also

set passthrough

display pppstats

Purpose

Displays status and activity information for a Point-to-Point Protocol (PPP) link, including SureLink statistics.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display pppstats
```

Examples

```
#> display pppstats
PPP Status (mobile0):
     state                : active
     ip address           : 166.130.0.157
     peer address         : 192.168.111.111
     primary dns addr     : 209.183.33.23
     secondary dns addr   : 209.183.33.23
     tx bytes             : 20403
     rx bytes             : 23430
     session tx bytes     : 250
     session rx bytes     : 736
     idle resets          : 41
     no carrier resets    : 0
     no service resets    : 0
     admin resets         : 0
     non-admin resets     : 2
     surelink resets      : 0
     lcp keepalive resets : 0
     last reset reason    : idle
     tx timer             : 0 seconds
     rx timer             : 3600 seconds
     session time         : 2 hours, 50 minutes, 15 seconds
     tx idle time         : 2 hours, 50 minutes, 15 seconds
     rx idle time         : 35 minutes, 41 seconds
     lcp echo requests    : inactive
     SureLink statistics:
       session successes              : 0
       session failures               : 0
       session consecutive failures   : 0
       session bypasses               : 0
       total successes                : 0
       total failures                 : 0
```

```
       total link down requests       : 0       total bypasses
: 0
```

Output

PPP status and activity Information returned by "display pppstats"

This information is specific to cellular-enabled Digi products.

display pppstats

Displays status and activity information for a PPP link and SureLink statistics.

In these status and activity values, a **session** is a PPP session. The session statistics are reset to zero at the start of a new PPP link. The **total** statistics are the accumulated totals for all sessions since the device booted.

state

The current state of the PPP link. **Active** indicates that a PPP link is up. **Inactive** means the PPP link is down. Inactive is indicated when the link is coming up, or going down.

ip address

The PPP WAN IP address of the Digi device. This is the IP address used to communicate over the cellular network. The carrier assigns this IP address most of the time, but can also be given to the network by the Digi device.

primary dns addr
secondary dns addr

These are addresses for the DNS nameservers used for performing name lookups. These DNS addresses are assigned by the carrier most of the time, but can also be assigned by users.

tx bytes
rx bytes

The total number of bytes transmitted (**tx**) or received (**rx**) over the PPP link since the last reboot.

session tx bytes
session rx bytes

The number of bytes transmitted (**tx**) or received (**rx**) over the PPP link in the current PPP session.

reset status

These PPP status values describe the reason for terminating the PPP link.

idle resets

The number of resets because the idle timeout was reached/exceeded for transmitted and received data. These idle timeouts are set by the **set ppp** command. Most of the time, no idle timeout is used on transmitted data.

no carrier resets

The number resets because the carrier was dropped for any reason.

no service resets

The number of resets because the data network was not available.

Note For Digi Cellular Family products, **no carrier** and **no service** resets indicate

problems with your cellular service. **No service** resets could be cased by low signal strength. Review the signal strength and reposition the antenna or Digi device as needed. An external high-gain or directional antenna may be needed. These resets can also be due to roaming issue. To check the signal strength and current carrier settings, issue a **display mobile** command.

admin resets

The number of resets done for administrative purposes, such as issuing "kill" commands, or disconnecting a PPP session in the web interface's Connections page by clicking the **Disconnect** button.

non-admin resets

The number of LCP termination requests made from the network; that is, the network notifies the Digi device that the PPP link is being brought down.

surelink resets

The number of resets caused by SureLink bringing down the PPP link and reestablishing it. SureLink performs three tests to monitor the integrity of the PPP link: ping, DNS, and TCP connection testing. The **set surelink** command has options for setting how these tests are performed (see set surelink). If SureLink is unable to complete these tests, it concludes that the link is broken, and reestablishes the connection.

lcp keepalive resets

The number of resets caused by the LCP keepalive tests. LCP keepalive tests are similar to the SureLink link integrity monitoring tests, and perform the equivalent of a ping test for the PPP link. If the cellular network does not ping back after the number of replies specified by the number of consecutive missed replaces on the **set pppoutbound** command option **lcp_ka_max_missed_replies**, the LCP keepalive feature drops the link.

last reset reason

The reason for the most recent reset of the PPP link.

idle

An idle reset brought down the link last.

lcp keepalive

An LCP keepalive reset brought down the link last.

surelink

Surelink tests failed and brought down the link.

no service

The modem received a **no service** indication on the monitoring channel, and brought down the link.

no carrier

The modem dropped the link (hard), and was responsible for the

termination of the last link.

administrative

A user issued a **kill** command, or disconnected the PPP link in the web interface's Connections page.

non-administrative

The network initiated closure of the last link.

unknown

The link was brought down for unknown reasons. This status is also displayed if the PPP connection has not been brought down since the Digi device was last rebooted. Since it is not possible for **last reset reason** information to persist across resets, this unknown state indicates that it is not clear which event may have been responsible for a reset that occurred at some point in the device's operation.

tx timer

The time, in seconds, after which if no data is transmitted, the PPP link is disconnected. Typically, this value is **0** (disabled).

rx timer

The time, in seconds, the PPP link disconnects if noo data is transmitted. An idle reset ends the PPP session and reestablishes it to prevent the carrier network from dropping an inactive call. The default is **1440** seconds (**24** minutes). This value is also known as the **Inactivity timeout** in the web interface's **Mobile Settings**.

session time

The duration of the current PPP session. To display total system uptime, issue a **display uptime** command.

rx idle time
tx idle time

The amount of time since data was last received (**rx**)/or transmitted (**tx**) by the Digi device.

lcp echo requests

The number of Link Control Protocol (LCP) echo requests that have been sent after a "quiet" interval, in order to test the PPP link and/or keep it alive. For Digi Cellular products, LCP echo requests are typically not used.

SureLink Statistics

Digi SureLink™ provides an "always-on" mobile network connection to ensure that a Digi Cellular Family device is in a state where it can connect to the network.

The statistics displayed for Digi SureLink pertain to the periodic tests, known as Link Integrity Monitoring tests, that are run over the established PPP connection to ensure that end-to-end communication is possible. There are three Link Integrity Monitoring tests available:

- Ping Test
- TCP Connection Test
- DNS Lookup Test.

For descriptions of these tests, see set surelink.

In the SureLink statistics, a **session** is a PPP session. The session statistics are reset to zero at the start of a new PPP connection. The **total** statistics are the accumulated totals for all sessions since

the device booted. The **tests** are the SureLink Link Integrity Monitoring tests that are configured to run when the mobile network connection is established.

session successes

The number of times a configured test attempts and succeeds in a PPP session.

session failures

The number of times a configured test attempts but fails in a PPP session.

session consecutive failures

The number of consecutive failures for a test. When a test is successful, the consecutive failures counter is reset to zero. The consecutive failures counter indicates a device's “progress” toward the configured maximum number of consecutive failures, after which the PPP link is taken down (and restarted).

session bypasses

If a configuration parameter is bad, a test is bypassed rather than considered to have succeeded or failed. This means the test was not run. If the PPP connection goes down while a test is in progress, that test may be classified as bypassed, since it could not be run.

Note The PPP link may go down for many reasons, independent of SureLink testing.

total successes

The total number of times a configured test attempts and succeeds since the Digi device was booted.

total failures

The total number of times a configured test attempts but fails since the Digi device was booted.

total link down requests

The number of times the SureLink feature has failed consecutively the configured number of failures and, as a result, requested that PPP shut down and restart its connection. This statistic counts such occurrences during the current device boot. SureLink itself does the PPP stop/start; it sends a message to PPP requesting a start/stop, due to a Surelink test failure.

total bypasses

The total test bypasses (see **session bypasses** statistic) since the Digi device was rebooted.

See also

- set ppp
- set surelink

display provisioning

Purpose

Displays the current provisioning parameters and other information in the Digi device's CDMA cellular module.

Before using the **provision** command to provision the CDMA module, Digi recommends using this command to determine which parameters are already set in the module.

Important: Close mobile PPP sessions before issuing provisioning commands

The **provision** and **display provisioning** commands cannot be used while mobile Point-to-Point Protocol (PPP) sessions are active. To close any existing mobile PPP sessions:

1. Disable the mobile PPP interface by entering a **set mobileppp** command with these options:

```
#> set mobileppp index=index of SIM card (1 or 2) state=disabled
```

2. Identify the ID of the mobile PPP session by issuing a **who** command. Any active PPP sessions are listed in the **Protocol** column as **ppp [connected]**. Note the ID number assigned to the PPP session. In the following example, the active PPP connection has a session ID of 2.

```
#> who

ID  From                  To                    Protocol
 Sessions
--  --------------------  --------------------  ---------------  ---
--
1   serial 1              local shell           term
2   166.130.0.159         166.130.0.154         ppp [connected]
3   166.130.0.159:57078   184.73.237.26:3197    dcloud tcp
4   10.8.16.115           local shell           telnet
```

3. Once you have identified the session, issue a **kill** command to end the mobile PPP session, specifying the ID for the mobile PPP session displayed in the **who** command, for example:

```
#> kill 2
```

4. Enter the **display provisioning** command (see the example output).

```
#> display provisioning
```

5. After displaying provisioning settings and optionally entering a **provision** command to change settings as needed, enable the mobile PPP interface by entering another **set mobileppp** command:

```
#> set mobileppp state=enabled
```

Required permissions

For Digi products with two or more users, you must set permissions to **set permissions display=execute** to use this command. In addition, the commands noted in the previous procedure, **set mobilepp**, **who**, and **kill**, have their own permissions, which are noted in their respective command descriptions. See set permissions for details on setting user permissions for commands.

Syntax

```
display provisioning
```

Examples

```
#> display provisioning

Current Provisioning Information:

MDN:                : 9149561099
MSID:               : 9143737012
MIP Profile:        : 1
NAI:                : acme06@sprintpcs.com
Home Address:       : 0.0.0.0
PRIHA:              : 255.255.255.255
SECHA:              : 68.28.89.78
HA SS:              : Set
HA SPI:             : 1234
AAA SS:             : Set
AAA SPI:            : 1234
RTUN:               : 1
```

See also

- display carriers
- display mobile
- provision
- set mobile

display proxyarp

Purpose

Displays the current contents of the proxy ARP table.

Use the proxy ARP table for troubleshooting only. There is no user interface to manage this table by explicit addition of removal of entries. This is primarily used in IP pass-through mode.

An entry in the proxy ARP table means that the Digi device responds to ARP requests for that IP address, as if the IP address were configured for the responding interface. This is useful in that the host making the ARP request forwards packets destined for that IP address to the Digi device, which then forwards them as the next routing hop.

Required permissions

For Digi products with two or more users, you must set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display proxyarp
```

See also

- set mobileppp: the **proxy_arp={on|off}** option.
- set passthrough: the **proxyarp={enabled|disabled}** option.
- set ppp: the **proxy_arp={on|off}** option.

display route

Purpose

Displays Route Table entries.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display route
```

Examples

```
#> display route

 Route Table:

Destination          Gw              Refs Mhome Iface    mtu  hops ttl
flags
 0.0.0.0/0           10.8.1.1        2    0     eth0     1500 1    INF
UGS
 10.8.0.0/16         *               1    0     eth0     1500 0    INF          UC
 127.0.0.0/8         127.0.0.1       1    0     LOOPBACK 1500 1    INF
UGJS
 127.0.0.1/32        127.0.0.1       4    0     LOOPBACK 1500 0    INF          UH
```

Output

Destination

The destination IP address or subnetwork for a routing table entry. **0.0.0.0** is the “default gateway” entry, meaning it is the entry used to reach a destination for which there is no other (more suitable) route. The value following the / (for example, **16** in **10.8.0.0/16**) specifies the subnet for that destination. **16** is a subnet mask of **255.255.0.0.**

Gw

The gateway IP address to which a packet is forwarded to reach the destination. This is the first “hop” if the destination is on a subnet that is not immediate to the Digi device. The gateway value ***** (asterisk) means that the destination can be reached by sending the packets out the interface associated with the route.

Mhome

Multihome index on the configured interface.

Iface

The name of the local network interface associated with the route.

mtu

The Maximum Transmit Unit (MTU) size of the packets that are sent through the interface associated with the route. This size includes all octets of the packet starting with the IP header, or more generally, the maximum number of octets that can follow the datalink header (for example, Ethernet, PPP, and so on).

hops

The number of “hops” (intermediate routers) between the Digi device and the destination. Also referred to as the “metric”. This value is generally not useful other than to indicate that the destination is on a subnet that is not local to the Digi device.

ttl

The Time To Live (TTL) for the IP packets. For each “hop” that a packet takes in being forwarded to its destination, the TTL value in its IP header is decremented by one. If the TTL reaches zero, the packet is discarded. This prevents routing loops where a packet is routed indefinitely among routers in an IP network (or the Internet). The value **INF** means that the lifetime of the IP packets is infinite.

flags

Routing flags that indicate the nature of the routing table entry.

Flag	Indication
U	Route is up. If not set, route is down.
H	Target is a host. If not set, target is a network
G	Use gateway. If not set, target is directly connected.
D	Dynamically installed by daemon or redirect. If not set, the routing entry is created statically.
J	Output through this route rejected
C	Entry can be cloned.
L	Cloned entry
S	Static entry configured by a user.

See also

set forwarding

display sadb

Purpose

Displays entries in the Security Association (SA) database. The SA database lists the connections to VPN servers. Each entry identifies the subnets traffic is being routed between, and the security protocols chosen for the VPN tunnel. This information can also be displayed with other Virtual Private Network (VPN) information.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display sadb
```

Examples

```
#> display sadb

 SADB Table:

Source IP      Dest. IP        Protect Mode  SPI    Hash  Enc   TTL-sec
TTL-kb
71.216.228.97  166.130.103.197  esp   tunnl   3206815678   md5   3des
216    INF
```

See also

- display ikesa
- display ipsecspd
- display spd
- display vpn
- set vpn
- vpn

display scancloak

Purpose

Displays current settings for the Network Port Scan Cloaking feature.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display scancloak
```

Examples

```
#> display scancloak

Network Port Scan Cloak Status:

  Cloak state: on

  Values configured in the network stack:

               Ping        TCP         UDP         DNS Proxy
    global     off         off         off         N/A
    eth0       off         off         off         off
    mobile0    off         on          on          on


Network Port Scan Cloak Statistics:

  Packets received but discarded due to cloaking:

               Ping        TCP         UDP         DNS Proxy
    global     0           12          4           0
    eth0       0           0           0           0
    mobile0    0           12          4           0
```

See also

To display all available network interfaces available on the Digi device and determine whether Network Port Scan Cloaking is desired on any of the interfaces, enter a **display_netdevice** command See display device.

- revert
- set scancloak
- show

display serial

Purpose

Displays serial modem signals (DTR, RTS, CTS, DSR, DCD).

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display serial
```

Examples

Serial modem signals for a device with one serial port

```
#> display serial

Serial Status:

  Port#    RTS    CTS    DSR    DCD    DTR     RI    OFC    IFC
     1     on    off    off    off    off    off     on    off
```

Serial modem signals for a device with two serial ports

```
#> display serial

Serial Status:

  Port#    RTS    CTS    DSR    DCD    DTR     RI    OFC    IFC
     1     on    off    off    off     on    off    off    off
     2     on    off    off    off     on    off    off    off
```

See also

- info serial
- set serial

display smscell

Purpose

Displays status and statistics of the Short Message Service (SMS) feature.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display smscell [general]
[users]
[scl]
[recvlog]
[recvlog=tail]
[sentlog]
[sentlog=tail]
[statistics]
[all]
```

Options

Note For the **recvlog** and **sentlog** options, the log is displayed.
If you specify the **tail** keyword, the command waits for new messages to be added to the log. As new messages are added to the log, they are displayed. Enter **Ctrl-C** to end the command.

The received and sent message logs contain only the most recent messages, not a full history of messages.

general

Display SMS general status.

users

Display SMS user information.

scl

Display the SMS sender control list.

recvlog

Display the received message log, reporting the most recent 10 messages received.

recvlog=tail

Display and “tail” the received message log.

sentlog

Display the sent message log, reporting the most recent 10 messages sent.

sentlog=tail

Display and “tail” the sent message log}.

statistics

Display SMS general statistics and counters.

all

Display all the above SMS status categories.

Examples

```
#> display smscell general

Cellular SMS General Status Information:

  SMS feature enabled: yes
  Password configured: no
  Default actions for received command messages:
    ACK all accepted received command messages: no
    NAK messages if password validation fails:  no
  Extended detail (verbose) SMS event logging:  no
  Default message receiver: 0x80000001 (logonly)

#>
#> display smscell users

Cellular SMS User (Command) Information:

  Number of user entries: 5
                                          Requires  Read  Read Hold   Ref
    User ID   Flags      User Name        Password  QLen  (Seconds)  Count
    --------  ---------  ---------------  --------  ----  ---------  -----
    80000001  0000 0111  _default_           no       0       600      1
    80000002  0000 0111  _builtincmd_        no       0       600      1
    80000003  0000 0011  _alarms_            no       0       600      1
    80000004  0000 1111  python              no       0       600      1
    80000005  0000 0111  cli                 no       0       600      1
    80000006  0000 0111  dcloud_msgsrv       no       0       600      1

    Flag descriptions:     If flag is set (1):
    ENABLED   .... ...1    This user entry is enabled.
    SEND      .... ..1.    This user can send messages.
    RECEIVE   .... .1..    This user can receive messages.
    READHOLD  .... 1...    Received messages are being held.
    RELEASED  .1.. ....    This user entry is released.
```

```
#> display smscell statistics

Cellular SMS General Statistics/Counters:

  Message send counters:
    User send requests                   : 37
    Send request successes               : 37
    Send request failures                : 0
    Send request bytes (unencoded)       : 962
    Messages using multiple SM blocks    : 2
    SM blocks queued (current)           : 0
    SM blocks queued (maximum)           : 2
    SM block submit requests             : 39
    SM block submit in progress          : 0
```

```
    SM block submit successes             : 39
    SM block submit failures              : 0
    SM block submit retries               : 0

  Message receive counters:
    Received short message (SM) blocks    : 52
    Received SM octets (decoded)          : 5401
    Received complete messages            : 37
    Messages requiring reassembly         : 15
    SM blocks in reassembly               : 0
    Duplicate SM blocks discarded         : 0
    SM blocks discarded in reassembly     : 0
    Messages discarded: SMS disabled      : 0
    Messages discarded: SCL rejection     : 0
    Messages discarded: password fail     : 0
    Messages discarded: read queue full   : 0
    Messages discarded: read hold timeout : 0
```

See also

- set smscell
- smscell
- show smscell

display sockets

Purpose

Displays information about how socket resources are being used by the system.

Sockets provice access to IP network connections, open devices, open files and XBee radio access.

A Digi device's operating system has 128 sockets available. If the system runs out of sockets, various features may not work correctly. Socket use is primarily dynamic. Some features acquire a socket and use it for a long time, while others get a socket, use it, and release it back to the system. The ability to display the sockets statistics is a troubleshooting aid, providing information on how many sockets are in use, allocated for which purpose, and how many are available.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display sockets
```

Examples

```
#> display sockets

Socket Status:

Total Sockets        : 128
Free Sockets         : 104
Used Sockets         : 24
  IP Network Sockets : 21
  Device Sockets     : 3
  File Sockets       : 0
  Mesh Sockets       : 0

Maximum TCP Sockets  : 128
Used TCP Sockets     : 14
```

See also

- display tcp
- display udp

display spd

Purpose

For Virtual Private Network (VPN) tunnels that have been configured, this displays both the IPSEC Security Policy Database (SPD) and Internet Key Exchange (IKE) table entries defined for VPN tunnels.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display spd
```

Examples

```
#> display spd

 IPSEC SPD Table:

 Idx, Selector(local ip:port,remote ip:port, protocol),Inner policy, Outer
policy,LEASTProtectRModePDHashRYEncRSTProtect Mode  Hash
  0 any, any, any                                   bypass
  1 192.0.0.0:0, 172.0.0.0:0, any                   esp     tunnl md5   3des
  2 192.0.0.0:0, 172.0.0.0:0, any                   esp     tunnl md5   3des
  3 any, 127.0.0.1:0, any                           bypass
  4 any, 192.168.0.0:0, any                         bypass
  5 any, 169.254.0.0:0, any                         bypass
  6 any, 166.130.103.197:0, any                     bypass


 IKE SPD Table:

 Source IP       Dest. IP        Hash  Enc   Auth    DH  Life(s)   Life(KB)
 166.130.103.197:71.216.228.97:0 md5   3des  psk     5   300       32768
```

See also

- display ikesa
- display ikespd
- display ipsecspd
- display sadb
- display vpn
- set vpn
- vpn

display tcp

Purpose

Displays active TCP sessions and active TCP listeners. **display tcp** provides information similar to that displayed by **netstat** commands on other operating systems.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display tcp
```

Examples

```
#> display tcp

 TCP Table:

Idx   family Recv-Q      Send-Q      LocalAddress          ForeignAddress Refs
State/Backlog RTO/Count
 8     IPV4   0/8192      0/8192      0.0.0.0:22            0.0.0.0:0  2
LISTEN/0      0/0
 9     IPV4   0/8192      0/8192      0.0.0.0:23            0.0.0.0:0  2
LISTEN/0      0/0
 21    IPV4   0/8192      0/8192      0.0.0.0:80            0.0.0.0:0  2
LISTEN/0      0/0
 10    IPV4   0/8192      0/8192      0.0.0.0:443           0.0.0.0:0  2
LISTEN/0      0/0
 17    IPV4   0/8192      0/8192      0.0.0.0:771           0.0.0.0:0   2
LISTEN/0      0/0
 11    IPV4   0/8192      0/8192      0.0.0.0:1027          0.0.0.0:0   2
LISTEN/0      0/0
 12    IPV4   0/8192      0/8192      0.0.0.0:2001          0.0.0.0:0   2
LISTEN/0      0/0
 14    IPV4   0/8192      0/8192      0.0.0.0:2101          0.0.0.0:0   2
LISTEN/0      0/0
 15    IPV4   0/8192      0/8192      0.0.0.0:2501          0.0.0.0:0   2
LISTEN/0      0/0
 16    IPV4   0/8192      0/8192      0.0.0.0:2601          0.0.0.0:0   2
LISTEN/0      0/0
 18    IPV4   0/8192      0/8192      0.0.0.0:50000         0.0.0.0:0   2
LISTEN/0      0/0
 41    IPV4   0/8760      0/8760      10.8.115.10:23        10.8.16.27:3579    7
ESTABLISHED   1000/0
 7     IPV4   0/8760      0/8760      127.0.0.1:41928       127.0.0.1:51089     6
ESTABLISHED   3000/0
 6     IPV4   0/8760      0/8760      127.0.0.1:51089       127.0.0.1:41928     6
ESTABLISHED   1000/0
```

See also

To display more TCP-related statistics, such as number of input and output bytes transmitted, issue an **info tcp** command. See info tcp.

- display sockets
- set network
- set profile
- set tcpserial

display techsupport

Purpose

Use this command when working with Digi Technical Support to investigate problems in a Digi device. It executes multiple **display**, **show**, and **set** commands to obtain device information.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display techsupport [interactive]
```

Options

interactive

Allows for interactively selecting which commands are executed. The commands are entered one at a time, output displayed, and a prompt displays which command will be executed next.

- To execute the next command, enter **c**.
- To skip the next command, enter **x**.
- To quit executing the sequence of commands, enter **q**.

Execute the commands in the following order:

```
display device
display versions
who
show panic a
display memory full
show serial
show profile
show network globalsettings if=*
show service
show snmp
show dnsproxy
show hostlist
display netdevice
display dnsserver
display sockets
display udp
display tcp
display route
display arp
display proxyarp
info ethernet if=*
info ip
info icmp
```

```
info tcp
info udp
show failover
display failover
show pppoutbound
show pppoutbound port=3
show pppoutbound port=5
show mobileppp
display pppstats
show passthrough
display passthrough
show surelink
display mobile
set module bootdiscovery=status
display gps
show ddns
display accesscontrol
display nat
show vpn all
display vpn
dhcpserver status
show mgmtglobal
show mgmtconnection
show mgmtnetwork
show devicesecurity
set python
display logging head=30
display logging tail=30
```

See also

You can also display the event log in the web interface by selecting **Management > Logging**.

- display logging
- set trace
- The Troubleshooting chapter of the *User Guide* for your Digi product.
- The Digi Support web page, to contact Technical Support, search Digi's knowledge base, ask a question on the Support forum, and get diagnostics and utilities.

display udp

Purpose

Displays current UDP listeners. **display udp** provides information similar to that displayed by **netstat** commands on other operating systems.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display udp
```

Examples

```
#> display udp

 UDP Table:

 Index  family  RecvQ       SendQ       LocalAddress          Refs
 3      IPV4    0/16384     0/16384     0.0.0.0:53            2
 25     IPV4    0/8192      0/8192      0.0.0.0:161           2
 20     IPV4    0/8192      0/8192      0.0.0.0:2362          3
 24     IPV4    0/8192      0/8192      0.0.0.0:60448         2
 4      IPV4    0/8192      0/8192      0.0.0.0:61445         3
 23     IPV4    0/8192      0/8192      0.0.0.0:62992         2
 2      IPV4    0/16384     0/16384     0.0.0.0:65024         2
```

See also

To display more UDP-related statistics, such as number of input and output bytes transmitted, issue the **info udp** command (see info udp).

- display sockets
- set udpserial

display uptime

Purpose

Displays the amount of time since the device was booted.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display uptime
```

Examples

```
#> display uptime

Uptime               : 1 day, 5 hours, 46 minutes, 41 seconds
```

display versions

Purpose

Displays Boot, POST, and EOS firmware version information and the Digi part numbers for each version.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display versions
```

Examples

```
#> display versions

ConnectPort X5 R:

Component           Part-Number     Release-Tag

BOOT                82002037        release_82002037_4P
POST                82002034        release_82002034_5P (V1.1.3)
Stored EOS          82002035        82002035_5P (V2.8.11)
Running EOS         82002035        82002035_5P (V2.8.11)
```

display vpn

Purpose

Displays all VPN-related status information, including Security Association (SA) database entries and Security Policy Database (SPD) entries.

Note This command applies to Digi Cellular Family products only.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display vpn
```

Examples

Display Virtual Private Network (VPN) status information

```
#> display vpn

SADB Table:

Source IP       Dest. IP        Protect Mode  SPI       Hash  Enc   TTL-sec
TTL-kb

IPSEC SPD Table:

Idx, Selector(local ip:port,remote ip:port, protocol),Inner policy, Outer policy,
Protect Mode  Hash  Enc   Protect Mode  Hash
  0 any, any, any
       bypass
  1 10.13.11.96:0, 0.0.0.0:0, any
       esp     tunnl sha1  aes
  2 10.13.11.96:0, 0.0.0.0:0, any
       esp     tunnl sha1  aes
  3 any, 127.0.0.1:0, any
       bypass
  4 any, 10.13.11.96:0, any
       bypass
  5 any, 166.203.137.88:0, any
       bypass

IKE SA Table:

Source IP       Dest. IP        Enc   Hash  Auth    DH  Life(s)   Life(KB)
66.165.177.14   66.165.177.14           nul 0         0
```

```
IKE SPD Table:

Source IP       Dest. IP        Hash  Enc   Auth    DH  Life(s)   Life(KB)
166.203.137.94:066.165.177.14:0 md5   aes   psk     2   240       32768
```

See also

- Commands that display VPN-related connection and status information:
  - display ikesa
  - display ikespd
  - display ipsecspd
  - display sadb
  - display spd
  - set vpn
- The VPN command (see vpn.
- The VPN settings in the web interface (**Network > Virtual Private Network (VPN) Settings**) and the online help for these settings.
- The *User Guide* section titled **Virtual Private Network (VPN) Settings**.

display vrrp

Purpose

Displays all Virtual Router Redundancy Protocol (VRRP) information, including all virtual routers that have been defined and their current status and state.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display vrrp
```

Examples

```
#> display vrrp
VRRP Status :
VRRP Virtual Router #1 (vrrp0) Status :
    state                     : disabled

VRRP Virtual Router #2 (vrrp1) Status :
s
    state                     : disabled
VRRP Virtual Router #3 (vrrp2) Status :
    state                     : disabled
 VRRP Virtual Router #4 (vrrp3) Status :

    state                     : disabled

VRRP Virtual Router #5 (vrrp4) Status :

    state                     : disabled

VRRP Virtual Router #6 (vrrp5) Status :

    state                     : disabled

VRRP Virtual Router #7 (vrrp6) Status :

    state                     : disabled

 VRRP Virtual Router #8 (vrrp7) Status :

    state                     : disabled
```

See also

set vrrp

display wimax

Purpose

Displays information for the WiMAX radio in the Digi device, including connection, network, and radio information, the network subscription list, RF activity, and statistics.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display wimax
```

Examples

```
#> display wimax

Connection Information:
  Radio Status: Connected
  Connection Duration: 13:23:31
  Disconnect Reason: Network Disconnect
  Subscription Name: Clear
  Network Type: Home
  NAP-ID: 000002
  RSSI: -78 dBm
  CINR: 12 dB
  Signal Quality: 3 of 5 bars
```

```
Network Information:
  IP Address: 184.78.91.72
  Gateway: 184.78.0.1
  Primary DNS: 66.233.235.12
  Secondary DNS: 75.94.255.12
```

```
Radio Information:
  Manufacturer: GCT Semiconductor, Inc.
  Model: Quanta WM553
  MAC Address: 00:17:C4:9C:53:F9
  SW Version: 1.10.1.2
  FW Version: 2.0.0.4
  HW Version: 0.0.7.0
```

```
Subscription List:
 INDEX  OPERATOR  NAME      NSP-ID  ACTIVATED
      1  Clear     Clear     000002  yes
      2  Clear     Sprint 4G  000002  yes
      3  Clear     Sprint PCS  000002  yes
```

```
Network List:
  NAME     TYPE     NAP-ID  RSSI  CINR
  Clear    Home     000002   -77    12
```

```
Neighbor List:
  BSID           FREQUENCY  PREAMBLE  RSSI  CINR
  000002:264FA2    2657000        40   -91     4
  000002:264FA3    2667000        72  -123   -10
  000002:2628FB    2551500        79   -94     1
  000002:264BE1    2647000        25  -123   -10
  000002:264BE2    2657000        57  -123   -10
  000002:264EEB    2667000        66  -123   -10
  000002:264F23    2667000        69  -123   -10
  000002:264F22    2647000        37  -123   -10
```

```
RF Information:
  BSID: 000002:264FA1
  UL PermBase: 8
  DL PermBase: 8
  Current preamble index: 8
  Previous preamble index: 8
  HO count: 0
  HO fail count: 0
  Resync count: 0
  HO signal latency: 83
  Combined CINR: 12 dB
  CINR: 9 dB
  CINR2: 9 dB
  Combined RSSI: -78 dBm
  RSSI: -80 dBm
  RSSI2: -83 dBm
  PER: 0.005193 [9/1733]
  Power control mode: 1
  TX power: 22 dBm
  TX power maximum: 23 dBm
  TX power headroom: 1 dBm
  UL burst data FEC scheme: QPSK (CTC) 1/2
  DL burst data FEC scheme: QPSK (CTC) 1/2
  UL burst data UIUC: 01
  DL burst data DIUC: 00
  Frequency: 2647000 KHz
  Power mode: Idle
```

```
Statistics:
  Session TX Bytes:  1552
  Session TX Frames: 11
  Session RX Bytes:  34053
  Session RX Frames: 442
  Total TX Bytes:    3210
  Total TX Frames:   23
  Total RX Bytes:    66475
  Total RX Frames:   741
  Connections: 2
  Connection Failed: 1
  Authentication Failed: 0
  User Requested: 0
  Network Disconnect: 1
  Radio Reset: 0
```

See also

- set wimax
- wimax

display wlan

Purpose

For Digi devices that are Wi-Fi-enabled, displays current wireless LAN parameters and operating status.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
display wlan
```

Examples

```
#> display wlan

Wireless LAN Status:

    state                : scanning
    channel              : 0
    SSID                 : ?
    BSSID                : 00:00:00:00:00:00
    tx rate              : 0 Mbps
    rx signal strength   : 68 %    [-49 dBm]
    authentication       : open system
    encryption           : open system
```

See also

- info wlan
- show and set wlan. The **show wlan** command displays additional wireless LAN information, including wireless LAN settings configured by **set wlan** and evaluations of the settings.
- status

display Xbee

Purpose

Displays a list of devices in an XBee network and information about the devices. You can a use this command to display information about a single specified device in an XBee network. Information displayed includes the node address and ID list, as well as individual node status. Options allow for refreshing, clearing, and sorting the list of displayed devices.

The settings returned by this command vary by XBee RF module and the XBee RF protocol running on the module. For complete descriptions of these settings, see the product manual for the XBee RF module.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

Display a list of devices in an XBee network

```
display xbee [refresh]
  [zigbee]
  [clear]
  [sort={id|network|extended|node|product}]
```

Display information for a specific device in an XBee network

```
display xbee [address={id|address}]
```

Options

refresh

Discover XBee devices using a Digi-specific method, the XBee **ND** command.

zigbee

Discover XBee devices using a ZigBee standard method known as **ZDO requests**.

clear

Clear list and discover devices again.

Note If no discover option (**refresh**, **zigbee**, or **clear**,) is specified on a **display xbee** command, the current list of XBee devices is displayed. New nodes are added to the list as they join the network.

sort={id|network|extended|node|product}]

Sort the displayed device list by the specified field.

id

Sort the device list by node ID. The node ID is a text label that can be set by the command **set xbee address=*node_id***.

network

Sort the device list by network address.

extended

Sort the device list by extended address.

node

Sort the device list by node ID.

product

Sort the device list by product type.

address={*id*|*address*}

The address of the node, specified by its node ID, 16-bit network address, or extended address. To display the node list, issue a **display xbee** command without options, or go to this page in the web interface: **Configuration > XBee Network.** If the address is not specified, the device settings for the local XBee module on the gateway are displayed.

id

PAN ID.

address

Extended address.

Examples

Display XBee network information

```
#> display xbee

XBee network device list

PAN ID:           0x1c1f - 0x460a9fcc95a2dc14
Channel:          0x13 (2445 MHz)
Gateway address:  00:13:a2:00:40:2d:c6:74!
Gateway firmware: 0x2141

Node ID         Network Extended address         Node type   Product type
--------------- ------- ------------------------ ----------- ------------
                [0000]! 00:13:a2:00:40:2d:c6:74! coordinator ConnectPort X5
1 coordinator

To display device details:
    display xbee address=(id|address)
#> display xbee

XBee network device list

PAN ID:           0x77e5 - 0x0000000000000072
```

```
Channel:          0x0e (2420 MHz)
Gateway address:  00:13:a2:00:40:31:8a:f3!
Gateway firmware: 0x2170

Node ID         Network Extended address         Node type   Product type

--------------- ------- ------------------------ ----------- ------------
                [0000]! 00:13:a2:00:40:31:8a:f3! coordinator X4 Gateway
                [f63f]! 00:13:a2:00:40:3c:54:a1! router      Digital IO Adapter

1 coordinator, 1 router
```

Display device details

```
#> display xbee address=00:13:a2:00:40:66:a1:b2!

Status of device: 00:13:a2:00:40:66:a1:b2!

            pan_id (OI): 0xd367
         ext_pan_id (OP): 0xf837cf1a74422f40
            channel (CH): 0xe
          net_addr (MY): 0x0
       association (AI): 0x0
  firmware_version (VR): 0x2170
  hardware_version (HV): 0x1e42
       device_type (DD): 0x30002
          children (NC): 10
       max_payload (NP): 255 (bytes)
    supply_voltage (%V): 3326 (mvolts)
       temperature (TP): 41 (degrees C)
              rssi (DB): 13 (-dBm)
          tx_power (PP): 18 (dBm)
      ack_failures (EA): 0
```

See also

- info xbee
- set xbee
- xbee
- The product manual for the XBee RF module available on Digi's Support site.

exit

Purpose

Terminates your current session.

Syntax

```
exit
```

Example

```
#> exit
```

See also

quit: The **quit** and **exit** commands perform the same operation.

findme

Purpose

Note This command is supported in Digi Connect ES only.

This command turns on/off the **Locator** LED to aid in finding a specific device among a group of devices. See the *Digi Connect ES Device Server Hardware Setup Guide* on Digi.com for the location of the **Locator** LED.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
findme [blink={on|off}]
```

Options

blink={on|off}

Causes the **Locator** LED to blink or stop blinking.

on

Causes the **Locator** LED to blink.

off

Causes the **Locator** LED to stop blinking.

Example

Display the current state of the Locator LED

```
#> findme
```

Cause the Locator LED to blink

```
#> findme blink=on
```

flashdrv

Purpose

Displays information about all types of flash drives that are attached to the ConnectPort X device.

The maximum size of the external flash drive supported for ConnectPort X4 devices is 4 GB.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
flashdrv
```

Options

None

Example

```
#> flashdrv

Volume      Bytes      Used        Available   Bad       Use%  Mounted on
            262.66 MB  254.03 MB   8.63 MB     0.00 MB   96.7  A:/
```

help and ?

Purpose

Displays help about a specific command.

Syntax

```
help [command]
```

OR

```
[command]?
```

Examples

```
#> help boot

syntax: boot [option]

options:

    action = (reset) {reboots device}
    action = (factory) {sets device settings to factory and reboots device}
    load = (host):(filename) {downloads new firmware}
#> boot?
```

The same command help as above is displayed.

```
#> help set serial

syntax: set serial [options...]

options:

    baudrate=(baudrate)
    databits=(5|6|7|8)
    stopbits=(1|2)
    parity=(none|odd|even|mark|space)
    flowcontrol=(none|software|hardware|custom)
    altpin=(on|off)
    closewait=(forever | 0-600)
    customflow=[rts|cts|dtr|dsr|dcd|ri|ixon|ixoff][,...]
    sigsonopen=(none|rtsdtr)

#> set serial?
```

The same command help as above is displayed.

See also

Displaying Online Help

info camera

Purpose

Displays statistical information about any camera attached to the Digi device.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-camera=read** or **set permissions s-camera=rw**. See set permissions for details on setting user permissions for commands.

Syntax

```
info camera
```

Example

```
#> info camera

Camera is connected.

Frames received:    0
Frames dropped:     0

Total bytes read:   0 (KB)
Average frame size: 0 (KB)
Largest frame:      0
Smallest frame:     0
```

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- show

- status
- set camera

info device

Purpose

Displays statistics from the device table. This information includes device-model information, MAC address, current Boot and POST code, firmware, memory usage, utilization, and uptime. The information displayed by this option is the same as that displayed by the **display device** command (see display device).

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info device
```

Example

```
#> info device

Device Information:

Product              : ConnectPort X4
MAC Address          : 00:40:9D:50:EA:E5
Firmware Version     : 2.14.1.5 (Version 82001536_K_SA5 09/30/2011)
Boot Version         : 1.1.3 (release_82001975_D)
Post Version         : 1.1.3 (release_82001753_G)
Product VPD Version  : release_82002003_B
Product ID           : 0x0081
Hardware Strapping   : 0x0044
CPU Utilization      : 8 %
Uptime               : 17 days, 1 hour, 25 minutes, 28 seconds
Current Date/Time    : Fri Mar 16 05:56:44 2007
Total Memory         : 33554432
Used Memory          : 13515372
Free Memory          : 20039060
Total Flash FileSys  : 3999744
Used Flash FileSys   : 363520
Free Flash FileSys   : 3636224
```

Output

Device statistics

Device information	Description
Product	The product name of the Digi device
Model	The model name of the Digi device.
MAC Address	A unique network identifier. All network devices are required to have their own unique MAC address. The MAC address is on a sticker on your Digi device. The number is displayed as 12 hexadecimal digits, usually starting with 00:40:9D.
Firmware Version	The current firmware version. This information may be used to help locate and download new firmware. Firmware updates may be downloaded from the Digi Support website.
Boot Version	The current boot version.
Post Version	The current POST version.
Product VPD Version	This field is a Digi part number beginning with “82” and its revision letter.
Product ID	A hexadecimal number that defines the “personality” of the printed circuit board/EOS (embedded operating system) combination.
Hardware Strapping	An ID that describes the hardware.
CPU Utilization	The amount of CPU resources being used by the Digi device.
Uptime	The amount of time the Digi device has been running since it was last powered on or rebooted.
Current Date/Time	The current date and time as set in the Digi device.
Total Memory	The total amount of memory (RAM) available.
Used Memory	The amount of memory (RAM) currently in use.
Free Memory	The amount of memory (RAM) currently not being used.
Total Flash FileSys	The Digi device has a flash file system for storing files, such as Python programs. This item shows the total amount of flash file system memory available.

Device information	Description
Used Flash FileSys	The amount of flash file system memory currently in use.
Free Flash FileSys	The amount of lash file system memory currently not being used.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- set system
- show
- status

info ethernet

Purpose

Displays Ethernet communications-related statistics.

For Digi devices with wireless LAN capability, this command displays two sets of statistics:

- Ethernet interface, labeled as **eth0**
- Wireless LAN interface, labeled as **wln0**

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-ethernet=read** or **set permissions s-ethernet=rw**. See set permissions for details on setting user permissions for commands.

Syntax

```
info ethernet
```

Examples

```
#> info ethernet

Ethernet statistics for eth0:

InBytes              : 263352167       OutBytes             : 1142750
InUcastPkts          : 15226           OutUcastPkts         : 14872
InNonUcastPkts       : 1145428         OutNonUcastPkts      : 519
InDiscards           : 0               OutDiscards          : 0
InErrors             : 0               OutErrors            : 0
InUnknownProtos      : 689098

#> info ethernet

Network interfaces:

Names InBytes   InUcastPkts InNonUcastPkts  OutBytes OutUcastPkts OutNonUcastPkt
----  ------- -----------   --------------  -------- ------------ --------------
eth0    2487755         511       31126       141980          428             17
wln0          0           0           0       335478            0            990
```

Output

Ethernet statistics

Statistic	Description
Names	The names of the Ethernet connections configured for the Digi device, including wireless LAN connections.
InBytes	Number of bytes received.
InUcastPkts	Number of Unicast packets received.
InNonUcastPkts	Number of non-Unicast packets received.
OutBytes	Number of bytes sent.
OutUcastPkts	Number of Unicast packets sent.
OutNonUcastPkts	Number of non-Unicast packets sent.
InDiscards	Number of incoming packets that were discarded.
OutDiscards	Number of outgoing packets that were discarded.
InErrors	Number of incoming packets that contained errors.
OutErrors	Number of outgoing packets that contained errors.
InUnknownProtos	Number of incoming packets where the protocol was unknown.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- show
- status
- set ethernet

info ia

Purpose

Displays statistics relating to Industrial Automation (IA).

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info ia
```

Example

```
#> info ia

IA Stats:

 critical errors                  :          0
 configuration errors             :          0
 resource errors                  :          0
 bad message errors               :          0
 timeout errors                   :          0
 discarded messages               :          0
 broadcast messages               :          0
 discarded broadcast messages     :          0
 disconnects                      :          0
 idle disconnects                 :          0
 kill disconnects                 :          0
 broadcasts                       :          0
 incoming requests                :          0
 outgoing requests                :          0
 incoming responses               :          0
 outgoing responses               :          0
```

See also

- **info** commands display statistical information about a device over time. In contrast, **display** commands focus display on real-time information, while the **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).

- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- show
- status
- set ia

info icmp

Purpose

Displays statistics related to Internet Control Message Protocol (ICMP) activity.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info icmp
```

Example

```
#> info icmp

ICMP statistics:

InMessages           : 10872          OutMessages          : 1926
InDestUnreachables   : 10872          OutDestUnreachables  : 1926
InErrors             : 0
```

Output

Statistic	Description
InMessages	Number of incoming messages.
InDestUnreachables	Number of incoming destination-unreachable messages received. A destination-unreachable message is sent to the originator when a datagram fails to reach its intended destination.
InErrors	Number of incoming received messages with errors.
OutMessages	Number of outgoing messages.

Statistic	Description
OutDestUnreachables	Number of destination-unreachable messages sent. A destination-unreachable message is sent to the originator when a datagram fails to reach its intended destination.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- ping: The **ping** command uses ICMP to ping an IP address.
- show
- status

info ip

Purpose

Displays statistics relating to Internet Protocol (IP) activity.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info ip
```

Example

```
#> info ip

IP statistics:

InReceives           : 97943           OutRequests          : 12615
InAddressErrors      : 15              DatagramsForwarded   : 0
InHeaderErrors       : 0               OutNoRoutes          : 1
InUnknownProtos      : 0               OutDiscards          : 0
InDiscards           : 0               FragCreates          : 0
NatPrivateToPublic   : 0               NatPublicToPrivate   : 0
ReassembleOks        : 0               FragOks              : 0
ReassembleFails      : 0               FragFails            : 0
AclExamines          : 0               AclAccepts           : 0
AclDiscards          : 0
```

Note The NAT-related fields are not displayed in products that do not include NAT support.

Output

Statistic	Description
InReceives	Number of datagrams received.

Statistic	Description
InAddressErrors	Number of received datagrams discarded because they were for another host and could not be forwarded.
InHeaderErrors	Number of received datagrams discarded because of invalid header information.
InUnknownProtos	Number of received datagrams discarded because the specified protocol is not available.
InDiscards	Number of received datagrams discarded for miscellaneous reasons.
ReassembleOks	Number of received datagrams that were successfully reassembled from fragments.
ReassembleFails	Number of received datagrams for which reassemby from fragments failed.
NatPrivateToPublic	Number of datagrams received from the private network, successfully translated by NAT, and returned to IP to be forwarded to the public network.
NatPublicToPrivate	Number of datagrams received from the public network, successfully translated by NAT, and returned to IP to be forwarded to the private network.
AclExamines	Number of received datagrams examined for access control filtering.
AclDiscards	Number of received datagrams discarded after being examined by access control filtering.
OutRequests	Number of datagrams given to IP to transmit.
DatagramsForwarded	Number of received datagrams forwarded to another host.
OutNoRoutes	Number of received datagrams discarded because no route to the destination IP address could be found.
OutDiscards	Number of outgoing datagrams that were discarded for miscellaneous reasons. This statistic is not used and is always zero.
FragCreates	Number of outgoing datagram fragments created.
FragOks	Number of outgoing datagrams that were fragmented.
FragFails	Number of outgoing datagram fragmentation attempts that failed. This statistic is not used and is always zero.
AclAccepts	Number of received datagrams accepted after being examined by access control filtering.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).

- boot to reset statistics.
- The **display** commands, to display current device status information for various features and functions.
- show
- status

info iridium

Purpose

For Digi devices that have an embedded Iridium® satellite modem, this command displays statistics associated with the Iridium satellite modem subsystem, including such elements as:

- Messages transmitted
- Message payload bytes transmitted
- Total serial bytes transmitted (includes message passing overhead and out-of-band messaging)
- Messages received
- Message payload bytes received
- Total serial bytes received (includes message passing overhead and out-of-band messaging)

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info iridium
```

Example

```
#> info iridium

Iridium Device Driver Serial Statistics:

  RX messages           : 0
  RX msg bytes          : 0
  RX bytes (total)      : 119389
  RX msg drops          : 0
  RX ring count         : 0
  TX messages           : 0
  TX msg bytes          : 0
  TX bytes (total)      : 171520
```

See also

- boot to reset statistics.
- display iridium

- Other **display** commands, to display current device status information for various features and functions.
- iridium
- set trace: The **iridium** option captures debugging information and error conditions from the Iridium satellite modem subsystem.

info orbcomm

Purpose

For Digi devices that have an embedded ORBCOMM® satellite modem, this command displays statistics for the satellite modem. The **state** setting, which shows the current state of the satellite modem, is always displayed. This command returns information of value only when the ORBCOMM device driver for the satellite modem is on.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. The satellite modem statistics are related to ORBCOMM messaging, and *may* provide insight into difficulties with and/or operation of the satellite link maintained by the modem. The device driver serial statistics are more interesting for Digi field diagnostics, as the numbers will change in unpredictable ways to most customers, depending on the operation of the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info orbcomm
```

Example

```
#> info orbcomm
Operating System Management of ORBCOMM Satellite Modem: disabled
#> info orb
Operating System Management of ORBCOMM Satellite Modem: enabled
ORBCOMM Satellite Modem Statistics:
  Packet errors        : 0
  SYNC count           : 0
  Acquire bursts       : 0
  COM bursts           : 0
  Reservation bursts   : 0
  Message TX count     : 0
  Globalgram TX count  : 0
  Report TX count      : 0
```

```
ORBCOMM Device Driver Serial Statistics:

  RX messages          : 8
  RX bytes (total)     : 749
  RX bytes dropped     : 0
  TX messages          : 16
  TX bytes             : 119
```

Output

Statistic	Description
Operating System Management of ORBCOMM® Satellite Modem	The current state of the ORBCOMM satellite modem, either enabled or disabled.
ORBCOMM Satellite Modem Statistics	Statistics from the ORBCOOMM satellite modem.
ORBCOMM Device Driver Serial Statistics	Serial statistics These statistics reflect raw serial communication, which may or may not have a direct correlation to logical operations being performed by the use. They are intended primarily for Digi diagnostic purposes.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display orbcomm
- **display** commands, to display current device status information for various features and functions.
- show
- status
- orbcomm
- set orbcomm
- set trace: The **orbcomm** option captures debugging information and error conditions from the ORBCOMM satellite modem subsystem.

info serial

Purpose

Displays statistics relating to serial-port communications activity.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info serial
```

Example

```
#> info serial

Serial port #1 statistics:

rbytes               : 423             tbytes               : 311
overrun errors       : 0               overflow errors      : 0
frame errors         : 0               parity errors        : 0
breaks               : 0

signal changes :      CTS     DSR      RI     DCD     RTS     DTR
                        3       0       0       0       1       1
```

Output

Statistic	Description
rbytes	Total data in: the number of bytes received.
overrun errors	The number of times FIFO has overrun. The next data character arrived before the hardware could move the previous character.
frame errors	The number of framing errors detected. The received data did not have a valid stop bit.

Statistic	Description
breaks	The number of break signals detected.
tbytes	Total data out: the number of bytes transmitted.
overflow errors	The number of times the Received buffer has overrun. The receive buffer was full when additional data was received.
parity errors	The number of parity errors detected. The received data did not have the correct parity setting
signal changes	For each signal (CTS, DSR, RI, DCD, RTS, DTR), the number of times the signal has changed states.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display serial
- Other **display** commands, to display current device status information for various features and functions.
- show
- status
- set serial

info tcp

Purpose

Displays statistics relating to Transmission Control Protocol (TCP) activity. Statistics displayed are those gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info tcp
```

Example

```
#> info tcp

TCP statistics:

InSegments           : 88056          OutSegments          : 44427
InErrors             : 74             RetransmitSegments   : 285
EstabResets          : 47             OutResets            : 99
PassiveOpens         : 126            ActiveOpens          : 59
Established          : 4              AttemptFails         : 99
```

Output

Statistic	Description
InSegments	Number of segments received.
InErrors	Number of segments received with errors.
EstabResets	Number of established connections that have been reset.
PassiveOpens	Number of passive opens. In a passive open, the Digi device server is listening for a connection request from a client.
Established	Number of established connections.
OutSegments	Number of segments sent.

Statistic	Description
RetransmitSegments	Number of segments retransmitted. Segments are retransmitted when the server doesn't respond to a packet sent by the client. This is to handle packets that might get lost or discarded somewhere in the network.
OutResets	Number of outgoing connections that have been reset.
ActiveOpens	Number of active opens. In an active open, the Digi device server is initiating a connection request with a server.
Attempt Fails	Number of failed connection attempts.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display tcp
- Other **display** commands, to display current device status information for various features and functions.
- set tcpserial
- show
- status

info time

Purpose

Displays statistics for SNTP clock sources that are configured using the **set clocksource** command.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info time
```

Example

```
#> info time

    Info Time:

      Time Source 2 (Ranking 11), SNTP Server:  time.nist.gov
      SNTP Requests       :          1     SNTP Good Responses :          0
      DNS Lookup Failures :          0     Send Failures       :          0
      Receive Failures    :          0     Malformed Responses :          0
      Receive Timeouts    :          0     Expired Responses   :          0


      Time Source 3 (Ranking 10), SNTP Server:  login.etherios.com
      SNTP Requests       :          3     SNTP Good Responses :          3
      DNS Lookup Failures :          0     Send Failures       :          0
      Receive Failures    :          0     Malformed Responses :          0
      Receive Timeouts    :          0     Expired Responses   :          0

  This report corresponds to these "clocksource" settings:

    #> set clocksource

      Idx  Source Type   State  Ranking  Interval  FQDN
      ---  ------------  -----  -------  --------  ----------------------
        1  cellular      on     60       n/a       n/a
        2  gps           off    20       n/a       n/a
        3  sntp          on     11       600       time.nist.gov
        4  sntp          on     10       600       login.etherios.com
        5  sntp          off    10       86400
           rtc           on     50       (for reference purposes only)
```

See also

- set clocksource
- set time
- set timemgmt
- boot to reset statistics.

info udp

Purpose

Displays statistics for User Datagram Protocol (UDP) communications activity.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info udp
```

Example

```
#> info udp
UDP statistics:
InDatagrams          : 46750          OutDatagrams         : 30755
InErrors             : 0              NoPorts              : 3062222
```

Output

Statistic	Description
InDatagrams	Number of datagrams received.
InErrors	Number of bad datagrams that were received. This number does not include the value contained by “No Ports”
OutDatagrams	Number of datagrams sent.
NoPorts	Number of received datagrams that were discarded because the specified port was invalid.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.

- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display udp
- Other **display** commands, to display current device status information for various features and functions.
- show
- status
- set udpserial

info wlan

Purpose

Displays statistics for wireless LAN (WLAN) communications activity.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info wlan
```

Example

```
#> info wlan

Wireless transmit statistics:

Bytes                : 2475685         Directed frames      : 45591
Broadcast frames     : 418             RTS frames           : 0
Retries              : 7827            Exceeded retry limit : 3
Broadcast errors     : 0               Not associated       : 7

Wireless receive statistics:

Bytes                : 851637330       Directed frames      : 125288
Broadcast frames     : 71876112        RTS frames           : 420
Retries              : 33933           No buffers           : 0
Invalid frames       : 42063           Duplicate frames     : 371
Exceeded lifetime    : 0               Decryption errors    : 6929736
Too large            : 0               Hardware overruns    : 0
Replay detected      : 0               MIC failed           : 0
```

Output

Wireless LAN (WLAN) statistics

The WLAN statistics may aid in troubleshooting network communication problems with your wireless network.

For additional wireless settings and an evaluation of the wireless settings, issue a **show wlan** command. See show.

Wireless transmit statistics

Statistic	Description
Bytes	Number of bytes transmitted.
Directed frames	Number of frames transmitted.
Broadcast frames	Number of broadcast frames transmitted.
RTS frames	Number of Request-to-Send (RTS) frames transmitted.
Retries	Number of times an outgoing frame is retransmitted because the acknowledgment for the frame was not received.
Exceeded retry limit	Number of outgoing frames that were dropped because the maximum number of retries were exceeded for the frame.
Broadcast errors	Number of broadcast frames dropped because the acknowledgment for the frame was not received.
Not associated	Number of outgoing packets dropped because the device had not yet associated with a wireless network.

Wireless receive statistics

Statistic	Description
Bytes	Number of bytes received.
Directed frames	Number of received frames.
Broadcast frames	Number of received broadcast frames.
RTS frames	Number of RTS frames received.
Retries	Number of incoming frames that have the retry bit set in their frame header. The retry bit indicates that the other side has attempted to transmit a given frame more than once.
No buffers	Number of received frames dropped due to no buffer.
Invalid frames	Number of incoming frames dropped because the frame appeared incorrect.
Duplicate frames	Number of incoming frames dropped because a given frame had already been received.
Exceeded lifetime	Number of fragmented frames dropped because the fragment timed out before the rest of the frame sequence was received.
Decryption errors	Number of frames dropped because they were not properly encrypted.
Too large	Number of frames dropped because their frame size was too big

Statistic	Description
Hardware overruns	Number of receive frames dropped because of hardware overruns.
Replay detected	Number of times an attempt to replay a packet was detected.
MIC failed	Number of times a data integrity check using Message Integrity Code (MIC) failed.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display wlan
- Other **display** commands, to display current device status information for various features and functions.
- show
- status
- set wlan

info xbee

Purpose

For Digi devices that have an XBee RF module, displays data counters that are specific to XBee sockets implemented using a Python application. Statistics displayed may aid in troubleshooting network communication problems with an XBee network.

Statistics that display are gathered since the statistic tables were last cleared by rebooting the Digi device, and include data, event, and error counts. These statistics are useful in understanding how the device is operating and can be helpful in finding problems. In particular, if an error counter is found to be increasing there may be a problem with the device.

To reset the statistics, reboot the device.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
info xbee
```

Example

```
#> info xbee
XBee sockets statistics:

Frames Sent                    : 0
Frames Received                : 0
Bytes Sent                     : 0
Bytes Received                 : 0
```

```
XBee sockets errors:

Transmit Frame Errors          : 0
Receive Frame Errors           : 0
Transmit Bytes Dropped         : 0
Receive Bytes Dropped by User  : 0
Receive Bytes Dropped by Stack : 0
```

```
XBee network statistics:

Frames received                : 41980
Bytes received                 : 157077
Frames transmitted             : 2
Bytes transmitted              : 0
Remote commands                : 2
Address discoveries            : 0
Route discoveries              : 0
Transmission retries           : 0
```

```
XBee network errors:

Removed from queue             : 0
Unable to transmit             : 0
Address not found              : 0
Route not found                : 0
Not acknowledged               : 0
No response or status          : 0
```

Output

Statistic/Error	Description
XBee sockets statistics	This section includes data counters that are specific to XBee sockets implemented using a Python application.
Frames Sent	The number of frames sent from local XBee device sockets.
Frames Received	The number of frames received by local XBee device sockets.
Bytes Sent	The total number of bytes sent from local XBee device sockets.
Bytes Received	The total number of bytes received from local XBee device sockets.
XBee sockets errors	This section includes error counters that are specific to XBee sockets implemented using a Python application. These values will help determine the quality of data that is being sent or received.
Transmit Frame Errors	The total number of frames that were not given to the XBee driver from the XBee device socket because of an internal error.
Receive Frame Errors	The total number of frames which were unable to be received by the XBee device socket because of an internal error.
Transmit Bytes Dropped	The total number of bytes dropped by XBee device sockets because of an internal error on transmission.
Receive Bytes Dropped by User	The total number of bytes dropped by the user because of an insufficiently sized receive buffer.
Receive Bytes Dropped by Stack	The total number of bytes dropped internally by XBee sockets because of insufficient internal buffers.
XBee network statistics	This section includes data counters for all activity on the XBee network.
Frames received	The total number of frames received.
Bytes received	The total number of bytes received.
Frames transmitted	The total number of frames transmitted.
Bytes transmitted	The total number of bytes transmitted.
Remote commands	The number of frames that were commands to remote nodes.

Statistic/Error	Description
Address discoveries	The number of frames that required the discovery of the network address of a remote node.
Route discoveries	The number of frames that required the discovery of the route to a remote node.
Transmission retries	The number of frames that were retransmitted because of they were not acknowledged by the remote node.
XBee network errors	This section includes error counts for all activity on the XBee network. These values help determine the quality of data that is being sent or received.
Removed from queue	The number of frames that could not be transmitted due to a time limit set by an application.
Unable to transmit	The number of frames that could not be transmitted due to a transmission error. This includes duty cycle limits and CCA and PHY errors.
Address not found	The number of transmitted frames for which the network address of a remote node could not be found.
Route not found	The number of transmitted frames for which the route to a remote node could not be found.
Not acknowledged	The number of transmitted frames that were not acknowledged by the remote node.
No response or status	The number of transmitted frames for which no indication of success or failure was received from the local radio.

See also

- **info** commands display statistical information about a device over time.
- **display** commands focus the display on real-time information.
- **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands).
- boot to reset statistics.
- display Xbee
- Other **display** commands, to display current device status information for various features and functions.
- show
- status
- set xbee
- xbee

- These ZigBee socket statistics are also available in the web interface; go to **Administration > System Information > XBee Network** and view the sections **Python Application XBee Socket Counters** and **Python Application XBee Socket Error Counts**.

iridium

Purpose

Powers on or off the Iridium® satellite modem. By default, when the Digi device is booted, the Iridium satellite modem is off. The Iridium satellite modem must be powered on either by this command or programmatically.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
iridium power={on|off}
```

Options

power={on|off}

Sets the power state of the Iridium satellite modem to on or off.

See also

- display iridium
- info iridium
- set dcloud_msgservice for details regarding Iridium-specific message-handling options.
- set trace: The **iridium** option captures debugging information and error conditions from the Iridium satellite modem subsystem.

kill

Purpose

Use the kill command to kill connections. The kill command is associated with the connections displayed by the **who** command.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
kill [range] [connection id]
```

Options

range

A range of connection IDs.

connection id

An ID for the connection.

Examples

Kill a connection on a specific port

```
#> kill 1F
```

Kill a connection on a range of ports

```
#> kill 1-3
```

See also

- close to close sessions created from the current connection.
- status to display the list of current sessions.
- who for information on determining active connections.

mobile_update

Purpose

Loads a preferred roaming list (PRL) into the cellular module on the Digi device. A PRL is a database that resides in a mobile device that contains information used during the system selection and acquisition process. It is built by the mobile service provider, and is normally not accessible to users.The PRL indicates which bands, sub bands and service provider identifiers will be scanned and in what priority order. Without a PRL, a mobile device may not be able to roam, or obtain service outside of the home area. There may be cases where missing or corrupt PRLs can lead to not having service at all.

On many networks, regularly updating the PRL is advised if the subscriber uses the device outside the home area frequently, particularly if they do so in multiple different areas. This allows the mobile device to choose the best roaming carriers, particularly “roaming partners” with whom the home carrier has a cost-saving roaming agreement, rather than using non-affiliated carriers. PRL files can also be used to identify home networks along with roaming partners, thus making the PRL an actual list that determines the total coverage of the subscriber, both home and roaming coverage.

The master subsidy lock (MSL) or a one-time subsidy lock (OTSL) associated with the module must be provided. The PRL file and the MSL or OTSL are obtained from the mobile service provider.

This command indicates whether the loading operation was successful. If it is unsuccessful, an error message is displayed.

Note This command applies to Digi cellular-enabled products that use the Sierra Wireless MC57xx series CDMA/EVDO modules only.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display settings, and **set permissions s-ppp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

```
mobile_update
  prl=tftp host IP address:prl file name
 msl=msl/otsl code
```

Options

prl=*tftp host ip address*:*prl file name*

The location and name of the PRL file to be loaded. ***tftp host ip address*** is the IP address of a TFTP server where the specified ***prl file name*** resides.

msl=*msl/otsl code*

The master subsidy lock (MSL) or a one-time subsidy lock (OTSL) associated with the module. This value is a six-digit activation or unlock code supplied by the mobile service provider.

Examples

```
#> mobile_update prl=192.168.1.1:90439.prl
```

See also

- display carriers
- display mobile
- provision
- set mobile

newpass

Purpose

Creates or changes user passwords for the device.

In Digi devices with a single-user model, changing the **root** user password also changes the password for Advanced Digi Discovery Protocol (ADDP). In Digi devices with the multi-user model, changing the **root** user password has no effect on ADDP. To change the ADDP password, specify **name=addp**.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions newpass=rw-self** for a user to set their own password, and **set permissions newpass=rw** to set another user's password. See set permissions for details on setting user permissions for commands.

Syntax

```
newpass [id=number|name=string]
```

Options

id=*number*

Specifies the ID of the user to be acted on.

name=*string*

Specifies the name of the user to be acted on.

Example

The **newpass** command initiates a dialog that changes the user's password.

User changing their own password

```
#> newpass
```

Changing another user's password

```
#> newpass name=jdoe
```

See also

- User Models and User Permissions in Digi devices
- set user for information on configuring users.

orbcomm

Purpose

Changes the state of the ORBCOMM satellite serial modem on the Digi device. This command controls state rather than particular settings associated with the modem). State changes take effect immediately and are not re-issued upon each startup (for instance).

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
orbcomm [power={on|off}]
  [load=tftp server:filename]
  [orbcomm action=factory]
```

Options

power={on|off}

Turns the power for the satellite modem on or off. For power-consumption reasons, the satellite modem always comes up in the **off** state, and must be powered on to be used. The command takes a few seconds to complete, as the running state of the modem is established.

load=*tftp server*:*filename*

Loads the specified firmware file on the satellite modem.

Note The **load** option can only be used if the satellite modem state is **off**. One of the reasons for this default behavior is to make it more difficult for a user to accidentally change the firmware. The syntax is similar to **boot load=** in that a TFTP host IP address and TFTP file name are paired with a colon separator as the value.

tftp server

The IP address of a host where the firmware file resides. The host must be running a TFTP server.

filename

The name of the firmware file.

orbcomm action=factory

Resets the ORBCOMM satellite modem to its internal default state. This reset operation clears any **locked** or **forced silent** state.

Example

```
#> orbcomm power=off
Powering off... complete.

#> orbcomm power=on
Powering on... complete.

#> orb load=10.8.110.22:m10_2006.ldr

  Retrieving firmware image... done.
  Image transfer to satellite modem progress:
           0 / 395110 bytes

error: update failed
```

See also

- display orbcomm
- info orbcomm
- revert: The **revert orbcomm** command reverts settings configured by **set orbcomm**.
- set orbcomm
- show: The **show orbcomm** command.
- set trace:The **orbcomm** option captures debugging information and error conditions from the ORBCOMM satellite modem subsystem.

ping

Purpose

Tests whether a host or other device is active and reachable.

To interrupt the **ping** command, enter **Ctrl-C**.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
ping ipaddress [options]
```

Options

ipaddress

Identifies the target of the **ping** command by its IP address.

options

The options associated with the **ping** command, which are:

count=0|**n**

The number of **ping** commands to be issued. **0** means ping until interrupted. The default is **0**.

interval=**milliseconds**

The ping time in milliseconds. The default is **1000** milliseconds.

size=**bytes**

The number of bytes to send in each ping packet. The value range is **1** through **1016** (bytes).The default is **56** bytes.

Examples

Specify a simple ping

The following command determines whether the specified host can be reached:

```
#> ping 199.150.150.10
PING 10.8.16.16: 64 data bytes
64 bytes from 199.150.150.10: icmp_seq=0 time=0 ms
64 bytes from 199.150.150.10: icmp_seq=1 time=0 ms
64 bytes from 199.150.150.10: icmp_seq=2 time=0 ms
64 bytes from 199.150.150.10: icmp_seq=3 time=0 ms
64 bytes from 199.150.150.10: icmp_seq=4 time=0 ms
[Ctrl-C entered to interrupt "ping" command]
--- 199.150.150.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

provision

Purpose

Provisions the CDMA module in a Digi device with cellular capability. Provisioning establishes configuration settings in the CDMA module for use in a mobile network. Examples of CDMA-based mobile service providers include Sprint, Verizon, Alltel, and Midwest. The CDMA module must be provisioned before you will be able to create a data connection to the mobile network.

Provisioning needs to be performed once only. It is not necessary for Digi Cellular Family devices that use GSM (Global System for Mobile Communication).

Provisioning is done either from the command line, using this command, or from the web interface, by launching the Mobile Device Provisioning Wizard from the Mobile Configuration page.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display provisioning settings, and **set permissions s-ppp=rw** to enter the "provision" command with options. See set permissions for details on setting user permissions for commands.

Provisioning types

There are several types of provisioning, each with different sets of parameters. Your mobile service provider can tell you which type is appropriate for your CDMA module.

- Simple IP only (SIPONLY).
- Mobile IP (MIP), which is a superset of SIPONLY.
- IP-Based Over-the-Air (IOTA).
- OTASP: Over-the-Air Service Provisioning (OTASP).

Check with your mobile service provider for provisioning parameters

The information that you need to specify during provisioning depends on your CDMA module and the settings that your mobile service provider has given you or already set up in your CDMA module.

Contact your mobile service provider for the most appropriate provisioning type and the required provisioning parameters. Have the ESN (Electronic Serial Number) for your Digi device ready to give to the provider. This number is located on the label on the bottom of the device.

Use "display provisioning" to get current provisioning parameters

You can query for the currently configured provisioning parameters in the CDMA cellular module by entering a **display provisioning** command.

Important: Close mobile PPP sessions before issuing provisioning commands

The **provision** and **display provisioning** commands cannot be used while mobile Point-to-Point Protocol (PPP) sessions are active. To close any existing mobile PPP sessions:

1. Disable the mobile PPP interface by entering a **set mobileppp** command with these options:

```
#> set mobileppp index=index of SIM card (1 or 2) state=disabled
```

2. Identify the ID of the mobile PPP session, by issuing a **who** command. Any active PPP sessions are listed in the **Protocol** column as **ppp [connected]**. Note the ID number assigned to the PPP session. In the example below, the active PPP connection has a session ID of **2**.

```
#> who

ID  From                  To                    Protocol
 Sessions
--  --------------------  --------------------  --------------------
 ------
1   serial 1              local shell           term
2   166.130.0.159         166.130.0.154         ppp [connected]
3   166.130.0.159:57078   184.73.237.26:3197    dcloud tcp
4   10.8.16.115           local shell           telnet
```

3. Once the session is identified, issue a **kill** command to end the mobile PPP session, specifying the ID for the mobile PPP session that was displayed in the **who** command, for example:

```
#> kill 2
```

4. Enter the **provision** command.

```
#> provision [command options]
```

5. Enable the mobile PPP interface by entering another **set mobileppp** command:

```
#> set mobileppp index=index of SIM card (1 or 2) state=enabled
```

Syntax

Depending on your mobile service provider and the information they have given you, parameters may be required, optional, or preset and not to be changed. Enter phone numbers as numbers only with no dashes; some mobile services providers do not accept dashes in phone numbers.

Display current provisioning parameters

```
display provisioning
```

Manually provision the module for a SIP-only network

```
provision type=siponly
  spc=service programming code (also known as master subsidy lock or MSL)
  mdn=mobile directory number
 min=mobile ID number
```

Manually provision the module for a MIP network

```
provision type=mip
  spc=service programming code (also known as master subsidy lock or MSL)
  mdn=mobile directory number
 min=mobile ID number
 nai=network access id
 aaass=AAA shared secret
 aaasstype={ascii|hex} default=ascii
  ha=home address
 priha=primary host agent IP address
 secha=secondary host agent IP address
 hass=host agent shared secret
 hasstype={ascii|hex} default=ascii
  haspi=index
  aaaspi=index
  rtun={0|1}
  profile=MIP profile number
```

Use IOTA to provision the module

```
provision type=iota
  spc=service programming code (MSL)(also known as master subsidy lock or MSL)
  mdn=mobile directory number
 min=mobile ID number
```

Use OTASP to provision the module

```
provision type=otasp
  otaspnumber=OTASP number - for example, *228 or *22899
```

Options

SIP-only provisioning parameters

type=siponly

Specifies that the CDMA module is being provisioned using the SIPONLY method.

spc=*service programming code* (MSL)

A six-digit number required to program CDMA module parameters. This code is also known as a master subsidy lock or MSL.

mdn=*mobile directory number*

The phone number of the CDMA module.

min=*mobile ID number*

How the CDMA module is identified in the cellular network. Depending on the cellular provider, this number may be the same as the mobile directory number.

MIP provisioning parameters

type=mip

Specifies that the CDMA module is being provisioned using the MIP method.

nai=*network access id*

Internet Authentication, Authorization and Accounting (AAA) protocols such as RADIUS or DIAMETER identify users with the Network Access Identifier (NAI). When used with Mobile IP and AAA, the NAI is composed of a username and a realm, separated by the **@** character. The username portion identifies the subscriber within the realm. The AAA nodes use the realm portion of the NAI to route AAA requests to the correct AAA server.

aaass=*AAA shared secret*

The shared secret used in authentication by Internet AAA protocols such as RADIUS or DIAMETER.

The format of the shared secret differs depending on whether it is entered in ASCII or hexadecimal, as specified by the **aaasstype** option.

If **aaasstype=ascii**, enter the shared secret as a string in quotation marks.

If **aaasstype=hex**, enter the shared secret as a hexadecimal number with no leading **0x** or trailing **h**.

aaasstype={ascii|hex} default=ascii

Specifies whether the AAA shared secret is specified in ASCII or hexadecimal form. This option affects how the shared-secret values are specified on the **aaass** option.

ha=*home address*

The home address for the CDMA module, specified as an IP address

priha=*primary host agent IP address*

The IP address of the primary host agent that provides mobile service for the CDMA module.

secha=*secondary host agent IP address*

The IP address of the secondary host agent that provides mobile service for the CDMA module.

hass=*host agent shared secret*

The shared secret used for authentication for the host agent.

The format of the shared secret differs depending on whether it is entered in ASCII or hexadecimal, as specified by the **hasstype** option.

If **hasstype=ascii**, enter the shared secret as a string in quotation marks.

If **hasstype=hex**, enter the shared secret as a hexadecimal number with no leading **0x** or trailing **h**.

hasstype={ascii|hex} default=ascii

Specifies whether the host agent shared secret is specified in ASCII or hexadecimal form. This option affects how the shared-secret values are specified on the **hass** option.

haspi=index
aaaspi=index

A Security Parameter Index (SPI) is an index identifying a security context between a pair of routers among the contexts available in the mobility security association. These are index options that set the security context between the host agent and the AAA server.

rtun= {0|1}

Enables or disables use of reverse tunnelling.

profile=*MIP profile number*

Specifies which of several profiles, or configuration scenarios, that the cellular module will use when communicating with the cellular network. This is a numeric value; the values available depend on your cellular provider.

IOTA provisioning parameters

type=iota

Specifies that the CDMA module is being provisioned using the IOTA method.

spc=*service programming code* (MSL)

A six-digit number required to program CDMA module parameters.

mdn=*mobile directory number*

The phone number of the CDMA module.

min=*mobile ID number*

How the CDMA module is identified in the cellular network. Depending on the cellular provider, this number may be the same as the mobile directory number.

OTASP provisioning parameters

type=otasp

Specifies that the CDMA module is being provisioned using the OTASP method.

otaspnumber=*OTASP number* for example, *228

A phone number for initiating an OTASP provisioning session. This number typically begins with ***228**, for example, ***22899**.

Example

```
#> provision type=otasp otaspnumber=*228
```

See also

- display mobile
- display provisioning to display the currently configured parameters in the CDMA cellular module.
- In the *User Guide* for your Digi product, the section on **Mobile Device Provisioning**. That discussion describes provisioning through a Wizard and the web interface, but the same concepts apply to command-line based provisioning.
- mobile_update
- set mobileppp
- who
- kill
- The Digi Support page for your Digi product. This page includes links to application guides and configuration and test documents for various mobile service providers.

python

Purpose

Manually executes a Python program from the command line. The **python** command is similar to a command executed on a PC. However, other than a program name and arguments for the program, the command takes no arguments itself, and is currently unable to spawn an interactive session.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions python-cmd=execute** for a user to use this command. The **python-files** permission is a separate permission that controls access to Python programs in the **Python** directory for the Digi device, but does not impact execution of this command. See set permissions for details on setting user permissions for commands.

Syntax

```
python [(TFTP server ip):]filename [program arguments...]
```

Options

[(*TFTP server ip*):]*filename*

The main file to be executed. This file can be either a file on the file system accessed through the Web UI, or a file accessible through a TFTP server on the network. This TFTP functionality reduces the number of times that you may need to place a program on the file system while developing and refining functionality. However, the TFTP behavior only works for the main program. Modules and packages must still be present on the file system to be used.

program arguments...

Arguments to be supplied to the program, as needed.

Example

```
#> python EmbeddedKitService.py
```

```
#> py dia.py
```

See also

- set python to manually execute a Python program.
- who: The **who** command can be used to view which Python threads are running. Python threads are listed for informational purposes. They cannot be killed with either the **kill** command or via the **Management > Connections** page in the web interface.

- The Digi Developer Community Wiki: this is a place to learn about developing solutions using Digi's communications portfolio, software and services, including Python, Device Cloud, DIA, and more.

  http://www.digi.com/wiki/developer/index.php/Main_Page

- Digi Python Custom Development Environment page:

  http://www.digi.com/technology/drop-in-networking/python.jsp

- The *Digi Python Programming Guide*. This guide introduces the Python programming language by showing how to create and run a simple Python program. It reviews Python modules, particularly modules with Digi-specific behavior. It describes how to load and run Python programs onto Digi devices, either through the command-line or web user interfaces, and how to run several sample Python programs. Find this guide on the Digi website at https://www.digi.com/wiki/developer/index.php/Digi_Python_Programmer's_Guide.
- The Python Support Forum on digi.com: Find answers to common questions and exchange ideas and examples with other members of the Digi Python development community at:

  http://www.digi.com/support/forum/forum.jspa?forumID=104

quit

Purpose

Use the quit command to log out of the device.

Syntax

```
quit
```

Example

```
#> quit
```

See also

exit.: The **quit** and **exit** commands perform the same operation.

reconnect

Purpose

Reestablishes a previously established connection; that is, a connection opened by a **connect**, **rlogin**, or **telnet** command. The default operation of this command is to reconnect to the last active session.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
reconnect [{serial port|p=serial port|s=session}]
```

Options

serial port

The serial port to which this command applies. Use this option to reconnect to a session opened by a connect command.

p=serial port | s=session

The serial port number or session number (displayed by the **status** command) to reconnect to.

Example

Reconnect to the last port used

```
#> reconnect
```

Reconnect to port 1

#> reconnect p=1

Reconnect to sessiosn 1

```
#> reconnect s=1
```

See also

- connect for information on establishing a connection on a selected port.
- close for information on ending a connection.
- status for information on gathering status on current connections.

- rlogin
- telnet

revert

Purpose

Sets a particular group of a devices' settings to its default values. Only one settings-group keyword can be specified per **revert** command. That is, entering several keywords on a single command to revert multiple settings is not allowed. A **revert all** command reverts all device settings but network, security, and host key/certificate settings.

Entering **revert user**, **revert group**, or **revert permissions**, displays a message indicating that those settings cannot be reverted individually, and instead must be reverted all together at the same time via the **revert auth** command. The **revert auth** command (revert authentication and authorization) reverts all users, all groups, and all permissions at the same time.

Required permissions

The only **set permissions** requirement is for the **revert all** command variant. Permissions used by the various **set** commands apply to the various **revert** command variants. **revert all** uses a different mechanism that bypasses the individual **set** commands, and therefore has its own permissions. To execute the **revert all** command, a user must have permissions set to **set permissions revert-all=execute**. See set permissions for details on setting user permissions for commands.

Syntax

```
revert [settings group option]
```

where [settings group option] is one of the following keywords:

```
[all|
  accesscontrol|
  alarm|
  auth|
  autoconnect [port=range]|
  buffer [port=range]|
  camera|
  clocksource [range=1-5]|
  dcloud_msgservice|
  ddns|
  devicesecurity|
  dhcpserver|
  dialserv [port=range]|
  dirp [range=1-20]|
  dnsproxy|
  ekahau|
  failover|
  forwarding|
  geofence [index=1-16]|
  gpio|
  host|
  hostlist|
  ia|
  login|
  mgmtconnection|
  mgmtglobal|
  mgmtnetwork|
  mobile [index=1-2]|
  mobileppp [index=1-2]|
  nat [instance=1-8]|
  network {all|globalsettings|interface=comma-separated list of interface
names]}|
  orbcomm|
  passthrough|
  pmodem [port=range]|
  position|
  ppp [port=1-5]|
  profile [port=range]|
  putty|
  python range=1-4|
  realport|
  scancloak|
  serial [port=range]|
  service|
  sharing [port=range]|
  slideshow|
  smscell{global|scl|python|command|all}
  snmp|
  socket_tunnel|
  surelink [index=1-2]|
  switches|
  system|
  tcpserial [port=range]|
  term [port=range]|
  trace|
  time|
  timemgmt
```

```
udpserial [port=range]|
video|
vncclient|
vpn {all|global|tunnel|phase1|phase2}|
vrrp [index=1-8]|
user (valid for single-user model devices only)
wimax|
wlan|
xbee
```

Options

all

Reverts all settings *except* these*:*

- Network settings
- Security settings (passwords and suppress login)
- Host key/certificate settings.

accesscontrol

Reverts the access control settings configured by the **set accesscontrol** command.

alarm

Reverts the alarm settings configured by the **set alarm** command.

auth

Reverts the permission settings configured by the **set permissions** command, the user settings configured by the **set user** command, and group settings, configured by the **set group** command.

This option is not available in Digi devices that implement the single-user model; see User Models and User Permissions in Digi devices for more information on user models.

autoconnect [port=*range*]

Reverts the Autoconnect settings configured by the **set autoconnect** command.

port=*range*

The serial port to which the autoconnect settings apply.

buffer [port=*range*]

Reverts the port-buffering settings configured by the **set buffer** command.

port=*range*

The port or ports to which the revert command applies.

camera

Reverts the camera settings configured by the **set camera** command.

clocksource [range=1-5]

Reverts the settings configured by the **set clocksource** command.

range=1-5

The time source entry for which settings are reverted.

idcloud_msgservice

Reverts the Device Cloud SMS settings configured by the **set dcloud_msgservice** command.

ddns

Reverts the Dynamic DNS (DDNS) settings configured by the **set ddns** command.

devicesecurity

Reverts the Device Cloud device security settings to their factory defaults.

dhcpserver

Reverts the DHCP server settings configured by the **set dhcpserver** command.

dialserv [port=*range*]

Reverts the Dialserv settings configured by the **set dialserv** command.

port=*range*

The port or ports for which the DialServ settings are reverted. If not specified, all settings are reverted.

dirp [range=1-20]

Reverts the device-initiated RealPort settings configured by the **set dirp** command.

range=1-20

Specifies which Device-Initiated RealPort connections are to be reverted. If not specified, all connections are reverted.

dnsproxy

Reverts the DNS proxy settings configured by the **set dnsproxy** command.

ekahau

Reverts the Ekahau client settings configured by the **set ekahau** command.

failover

Reverts the IP network failover settings configured by the **set failover** command.

forwarding

Reverts the port-forwarding settings configured by the **set forwarding** command.

geofence [index=1-16]

Reverts the geofence/GPS settings configured by the **set geofence** command.

index=1-16

The index number for the geofence. If not specified, all geofence settings are reverted.

gpio

Reverts the GPIO settings configured by the **set gpio** command.

host

Reverts the host name set by the **set host** command.

hostlist

Reverts the host settings configured by the **set hostlist** command.

ia

Reverts the Industrial Automation (IA) settings configured by the **set ia** command.

login

Reverts the login settings configured by the **set login** command.

mgmtconnection

Reverts the Device Cloud connection settings configured by the **set mgmtconnection** command.

mgmtglobal

Reverts the Device Cloud global settings configured by the **set mgmtglobal** command.

mgmtnetwork

Reverts the Device Cloud network settings configured by the **set mgmtnetwork** command.

mobile [index=1-2]

Reverts the mobile settings configured by the **set mobile** command.

index=1-2

The slot number of the SIM card. Required on dual-SIM devices. Enter **set mobile** without options to display these index numbers and primary and secondary SIM card designations.

mobileppp [index=1-2]

Reverts the mobile PPP connection settings configured by the **set mobileppp** command.

index=1-2

The index number of the SIM used for the mobile PPP connection. Enter **set mobileppp** without options to display these index numbers and primary and secondary SIM card designations.

nat [instance=1-8]

Reverts the Network Address Translation (NAT) and port/protocol forwarding settings configured by the **set nat** command.

instance=1-8

The NAT instance to which the **revert** command applies. If an instance is not specified, all instances will be reverted.

network {all|globalsettings|interface=*comma separated list of interface names*]}

Reverts these settings:

- Ethernet settings, configured by the **set ethernet** command.
- Network settings, configured by the **set network** command.
- Wireless configuration settings, configured by the **set wlan** command.

revert network all

Reverts settings for all interfaces to default settings.

revert network globalsettings

Reverts the global network settings. See set network for details on these options.

Note The global settings include the TCP options. Reverting the TCP options does not affect existing connections. TCP service listeners continue to use the previous option values until the service is restarted or a reboot is performed.

revert network interface=*comma-separated list of interface names*

Reverts the interface-specific network settings for the specified interfaces. The **interface** option can be abbreviated as **if**.

revert network

revert network with no options is equivalent to **revert network all**.

orbcomm

Reverts the ORBCOMM satellite settings configured by the **set orbcomm** command.

passthrough

Reverts the IP pass-through settings configured by the **set passthrough** command.

pmodem [port=*range]*

Reverts the modem emulation settings, configured by the **set pmodem** command.

[port=*range*]

The serial port for which modem emulation settings are reverted. Optional on a single-port device.

position

Reverts the position settings configured by the **set position** command.

ppp [port=1-5]

Reverts the Point-to-Point Protocol (PPP) outbound connection settings, configured by the **set ppp** command.

port=1-5

The physical interface to which the revert command applies.

profile [port=*range*]

Reverts the profile settings configured by the **set profile** command.

port=*range*

The serial port number or range of serial ports for which profile settings are reverted.

putty

Reverts the terminal emulation settings configured by the **set putty** command.

python [range=1-4]

Reverts the Python program settings configured by the **set_python** command.

range=1-4

The index or indices of the Python instance to be reverted. To view the indices, enter **set python** with no options.

realport

Reverts the Realport settings configured by the **set realport** command.

scancloak

Reverts the network port scan cloaking settings configured by the **set scancloak** command.

serial [port=range]

Reverts the serial, RCI serial, and RTS toggle settings configured by the s**et serial**, **set rciserial**, and **set rtstoggle** commands.

port=range

The serial port for which settings are to be reverted.

service

Reverts the service settings configured by the **set service** command.

sharing [port=*range*]

Reverts the port-sharing settings configured by the **set sharing** command.

[port=*range*]

The serial port for which port-sharing settings are to be reverted.

slideshow

Reverts the slideshow feature for displaying images from a USB storage device configured by the **set slideshow** command.

smscell {global|scl|python|command|all}

Reverts the SMS settings configured by the **set smscell** command. One or more settings groups can be specified. If no group is specified, **all** is assumed. Settings groups are:

global

Revert global SMS settings.

scl

Revert Sender Control List SMS settings.

python

Revert Python SMS settings.

command

Revert command SMS settings.

all

Revert all SMS settings (default).

snmp

Reverts the SNMP settings configured by the **set snmp** command.

socket_tunnel

Reverts the socket tunnel settings configured by the **set socket_tunnel** command.

surelink [index=1-2]

Reverts the Digi SureLink™ settings configured by the **set surelink** command.

index=1-2

The index number of the SIM to which the hardware reset threshold settings apply. Enter **set surelink** without options to display these index numbers and primary and secondary SIM card designations.

switches

Reverts the Multiple Electrical Interface (MEI) switch settings configured by the **set switches** command.

system

Reverts the system settings configured by the **set system** command.

tcpserial [port=*range*]

Reverts the TCP serial settings configured by the **set tcpserial** command.

port=*range*

The serial port for which TCP serial settings are to be reverted.

term [port=*range*]

Reverts the terminal connection settings configured by the **set term** command.

port=*range*

The serial port for which terminal connection settings are to be reverted.

time

Reverts the time settings configured by the **set time** command.

timemgmt

Reverts the time management settings configured by the **set timemgmt** command.

trace

Reverts the trace settings configured by the **set trace** command.

udpserial [port=range]

Reverts the UDP serial settings configured by the **set udpserial** command.

port=*range*

The serial port for which UDP serial settings are to be reverted.

user

Reverts the user settings configured by the **set user** command.

This option is valid for Digi devices that implement the single-user model only. See "User Models and User Permissions in Digi devices" on page 13 for more information on user models.The two-or-more-user model reverts users via the **revert auth** command.

video

Reverts the video settings configured by the **set video** command.

vncclient [port=*range*]

Reverts the settings configured by the **set vncclient** command.

port=*range*

The serial port for which VNC client settings are to be reverted.

vpn {all|global|tunnel|phase1|phase2}

Reverts the Virtual Private Network (VPN) settings configured by the **set vpn** command. Keyword options allow for reverting all or selected VPN settings. See set vpn for descriptions of the settings.

all

Reverts all VPN settings.

global

Reverts global VPN options.

tunnel

Reverts VPN tunnel settings.

phase1

Reverts Internet Key Exchange (IKE)/Internet Security Association and Key Management Protocol (ISAKMP) Security Association (SA) Phase 1 options.

phase2

Reverts IKE/ISAKMP SA Phase 2 options.

vrrp [index=1-8]

Reverts the Virtual Router Redundancy Protocol (VRRP) settings configured by the **set vrrp** command.

index=1-8

The index number for the virtual router for which settings are to be reverted. Enter a **set vrrp** command without options to display index numbers.

wimax

Reverts the WiMAX radio settings configured by the **set wimax** command.

wlan

Reverts the wireless settings configured by the **set wlan** command.

xbee

Reverts the XBee RF module settings configured by the **set xbee** command.

Examples

Reset a device's serial setting

The device serial setting is reset to the default serial configuration.

```
#> revert serial
```

Reset a serial port to default settings

```
#> revert serial port=2
```

See also

- boot
- The various **set** commands referenced in the **revert** command description.
- show

rlogin

Purpose

Performs a login to a remote system, also referred to as an rlogin.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
rlogin [esc=(char)] [{user=user name|-l user name}]
 [ip address]
```

Options

esc

A different escape character than the **~** (tilde) character used for the current Rlogin session. This character suspends a session from the remote host to return to the device server command line.

user=*user name* | -l *user name*

The user name to use on the remote system. If you do not specify a name, your device server user name will be used. The **-l *user-name*** option is for compatibility with the UNIX **rlogin** command.

ip address

The IP address of the system to which you are performing the remote login.

Examples

```
#> rlogin 10.0.0.1
```

See also

- telnet
- connect
- status
- close

send

Purpose

Sends a Telnet control command or special-character sequences when connected using the Telnet client.

Required permissions

For Digi products with two or more users, set permissions to **set permissions display=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
send {ao|ayt|brk|ec|el|escape|ga|ip|nop|synch}
```

Options

ao

Sends the **abort output** signal to discard output buffered on the peer.

ayt

Sends the **are you there** signal to test whether a host is still active.

brk

Sends the **break** signal to interrupt the executing application.

ec

Sends the **erase character** to delete the previous character.

el

Sends the **erase line** signal to delete the entire current line.

escape

Sends the **escape** character."

ga

Sends the **go ahead** signal.

ip

Sends the **interrupt process** signal to terminate the program running on the peer.

nop

Sends the **no operation** signal to the peer.

synch

Sends the **synchronize process** signal to the peer.

Examples

Send an “interrupt process” signal

```
#> send ip
```

Send an “are you there” signal

```
#> send ayt
```

See also

telnet for information on establishing Telnet sessions.

set accesscontrol

Purpose

Used to specify information that limits network access to this device, or display current access-control settings. For the Digi Connect WAN, the access-control settings also limit routing of packets through the device.

Required permissions

For Digi products with two or more users permissions must be set to
set permissions s-accesscontrol=read to display the access control settings, or
set permissions s-accesscontrol=rw to display and configure access control settings.

Syntax

Configure access control settings

```
set accesscontrol [enabled={on|off}]
  [autoaddsubnets={on|off}]
  [addrip[1-64]=ipaddress]
  [subnip[1-32]=ipaddress]
  [subnmask[1-32]=mask]
```

Display current access-control settings

```
set accesscontrol
```

Options

enabled={on|off}

Used to enable access control.

Take care when using this option, as improper settings can render the device inaccessible from the network. Specifically, setting the **enabled=on** without specifying any **addrip** option values disables all access.

on

Enables access control.

off

Disables access control.

autoaddsubnets={on|off}

Used to enable the automatic adding of subnets and subnet masks to this table. The IP subnets for the device server's network interfaces (Ethernet and PPP), may be automatically added to the table. This permits access by all IP sources on the device server's networks, without having to explicitly identify either the subnet IP addresses (and netmasks) or individual IP addresses.

on

Enables automatic adding of subnets and subnet masks.

off

Disables automatic adding of subnets and subnet masks.

addrip[1-64]=*ipaddress*

Up to 64 individual IP addresses that are allowed to access this device.

subnip[1-32]=*ipaddress*

Up to 32 subnet IP addresses. Any IP address in these subnets will be allowed to access this device server.

subnmask[1-32]=*mask*

A subnet mask associated with one of the 32 subnet IP addresses.

Examples

Set access control settings

```
#> set accesscontrol enabled=on addrip1=143.191.1.228
```

Set access control for a specific subnet

This command allows any IP address in the subnet **143.191.2.0 (**netmask **255.255.255.0**) to access this Digi device:

```
#> set accesscontrol enabled=on subnip1=143.191.2.0 subnmask1=255.255.255.0
```

Display access control settings

```
#> set accesscontrol
```

See also

- revert: The **revert accesscontrol** command reverts the settings configured by this command.
- show. The **show accesscontrol** command shows the current access-control settings in a Digi device.

set alarm

Purpose

Configures device alarms and display current alarm settings. Device alarms are used to send emails or SNMP traps when certain device events occur. These events include

- Data patterns detected in the serial stream.
- Changes in GPIO pin states.
- For cellular-enabled products, the average signal strength falling below a specified level for a specified amount of time, and the amount of cellular traffic for a specified period of time.
- Mobile temperature exceeding certain thresholds.
- Configuration changes to settings associated with the mobile device.

For Digi devices managed by a remote manager, the **cwm** option sends all alarms to a Device Cloud server.

The number of alarms that can be configured varies by product. To determine the number of alarms, enter **set alarm ?** and view the range specified for the **range** parameter. For example, this **range** parameter indicates that up to 8 alarms can be configured:

```
range=1-8
```

To avoid false errors, configure alarms while alarms are disabled, by entering a **set alarm state=off** command, then enable alarms after they are fully configured by entering **set alarm state=on**.

Required permissions

For Digi products with two or more users, set permissions to **set permissions s-alarm=read** to display current alarm settings, and to **set permissions s-alarm=rw** to display alarm settings and configure alarms. See set permissions for details on setting user permissions for commands.

Syntax

Configure alarms with general options—these settings apply to all alarms

```
set alarm [state={on|off}]
  [mailserverip=ipaddress]
  [from=string]
  [dcloud={on|off}]
```

Configure alarms for a range—set multiple alarms

```
set alarm range=range
  [active={on|off}]
  [type={email|snmptrap|sms|email_snmp|email_sms|snmp_sms|all}]
  [to=string]
  [cc=string]
  [subject=string]
  [priority={normal|high}]
```

```
  [sms_to=address]
  [sms_cc=address]
  [mode={match|gpio|rssi_gsm|rssi_1xrtt|rssi_1xevdo|ecio_1xrtt|
    ecio_1xevdo|cell_data|mobile_temp|1xevdo_unavail|
    mobile_unavail|mobile_cfg_chng}]
```

Configure alarms based on GPIO pin states, where n is the GPO pin number

```
set alarm range={1-32} mode=gpio
  [pins=list of pins]
  [highpins=list_of highpins]
  [lowpins=list of lowpins]
 [pin{n}={{high|1}|{low|0}}{ignore|x}}]
  [trigger_interval=seconds]
  [reminder={on|off}]
  [reminder_interval=seconds]
```

Configure alarms based on data pattern matching

```
set alarm mode=match
  match=string
 port=port_range
```

Note The **port=*port_range*** option is present only if there is more than one port on the device. ***port_range*** is replaced by the number of serial ports.

Configure alarms based on mobile signal strength—GSM

```
set alarm mode=rssi_gsm
  sig_strength_threshold=threshold
 time=time
 optimal_alarms_enabled={on|off}
```

Configure alarms based on signal strength—CDMA

```
set alarm mode={rssi_1xrtt|rssi_1xevdo|ecio_1xrtt|ecio_1xevdo}
  sig_strength_threshold=threshold
 time=time
  optimal_alarms_enabled={on|off}
```

Configure alarms based on 1xevdo service unavailable

```
set alarm mode=1xevdo_unavail
  time=time
 optimal_alarms_enabled={on|off}
```

Configure alarms based on mobile service unavailable

```
set alarm mode=mobile_unavail
  time=time
 optimal_alarms_enabled={on|off}
```

Configure alarms based on mobile configuration changes

```
set alarm mode=mobile_cfg_chng
```

Configure alarms based on mobile temperature changes

```
set alarm mode=mobile_temp
  temperature=temperature
 optimal_alarms_enabled={on|off}
```

Set alarms based on cellular data traffic

```
set alarm mode=cell_data
  cell_data=byte count threshold
 cell_data_type={receive|transmit|total}
  time=max time
```

Display current alarm settings

```
set alarm [range={1-8}]
```

Options

General alarm options

state= {on|off}

Enables or disables all alarms.

on

Enables all alarms.

off

Disables all alarms.

To avoid false errors, it is recommended that you configure alarms while alarms are disabled, and enable alarms after they are fully configured.

The default is **off**.

mailserverip=ipaddress

Used to configure IP address of the mail server to which alarm-triggered emails are sent.

from=*string*

The text to be included in the **from** field of an alarm-triggered email.

dcloud={on|off}

This option is displayed but not supported. For Device Cloud alarm functionality, see the alarms section of the *Device Cloud User Guide*.

Enables or disables sending of alarm notifications to the Device Cloud server.

on

Send all alarm notifications to the Device Cloud server. Turn this option on if your Digi device is managed by a Device Cloud server. Enabling this option is useful because it allows all alarms to be monitored from one location, Device Cloud. Enabling this option also allows Digi devices to send alarms to clients that would otherwise be unreachable from the Digi device, either because the Digi device is behind a firewall or not on the same network as the alarm destination.

off

Disables sending of alarm notifications to Device Cloud. Leave this option off if you do not manage your devices with Device Cloud or if you wish to have alarms sent from the device, for example, because an SNMP trap destination is local to the device, not Device Cloud.

Options for setting multiple alarms with the "range" option

range=*range*

The alarm or range of alarms for which alarm options are set. The number of alarms that can be configured varies by product. To determine the number of alarms, enter **set alarm ?** and view the range specified for the **range** parameter. For example, **range=(1 - 8)** indicates that up to 8 alarms can be configured.

active={on|off}

Enables or disables an alarm.

on

Enables an alarm.

off

Enables an alarm.

The default is **off**.

type={email|snmptrap|sms|email_snmp|email_sms|snmp_sms|all}

Used to determine what kind of an alarm is sent.

For SNMP traps to be sent, the IP address of the system to which traps are sent must be configured, by issuing a **set snmp** command with the **trapdestip** option. See "set snmp" on page 379.

email

Alarm is sent as an email.

snmptrap

Alarm is sent as an SNMP trap. If specified, the **subject** text is sent with the alarm. The MIBs for these traps are **DIGI-SERIAL-ALARM-TRAPS.mib**, and **DIGI-MOBILETRAPS.mib**.

sms

Alarm is sent as SMS text.

email_snmp

Alarm is sent as email and SNMP trap.

email_sms

Alarm is sent as email and SMS text.

snmp_sms

Alarm is sent as an SNMP trap and SMS text.

all

Alarm is sent as email, SNMP trap, and SMS text.

The default is **email**.

Note on sending alarms as SMS text: For SMS alarms, such alarms are suitable only for cellular products that support SMS, and only if SMS is included in the mobile service account for the cellular modem. There may be additional charges for SMS service or individual short messages, according to the customer's service agreement.

to=*string*

The text to be included in the **to** field of an alarm-triggered email.

cc=*string*

The text to be included in the **cc** field of an alarm triggered email.

subject=*string*

If **type=email**, this option specifies the text to be included in the **subject** field of an alarm-triggered email. If **type=snmptrap**, this option specifies the text to be included in the **Serial Alarm Subject** field of an alarm-triggered SNMP trap.

priority={normal|high}

The priority of the triggered email.

normal

The email is sent with normal priority.

high

The email is sent with high priority.

The default is **normal**.

sms_to=*address*

The address of the recipient to receive the SMS text. This option's value is typically specified as a destination phone number or a short code appropriate to SMS.

sms_cc=*address*

The address of recipient to receive a copy of the SMS text. This option's value is typically specified as a destination phone number or a short code appropriate to SMS.

mode={match|gpio|rssi_gsm|rssi_1xrtt|rssi_1xevdo|ecio_1xrtt|ecio_1xevdo|cell_data|mobile_temp|
1xevdo_unavail|mobile_unavail|mobile_cfg_chng}

The alarm mode, which determines what type of event will trigger an alarm.

The default mode is **gpio**, unless the Digi product does not support GPIO pins, in which case, the default is **match**.

match

An alarm will be triggered when a pattern is found in the stream of serial data on the specified port.

gpio

Transitions for GPIO pins will trigger alarms. See "GPIO pin state-based alarm options" on page 174 for more information.

rssi_gsm

Triggers alarms when the average signal strength on a GSM device falls below a specified threshold for a specified amount of time. Optionally, a subsequent alarm is triggered when signal strength returns to the optimal state above the threshold.

rssi_1xrtt

Triggers alarms when the average RSSI 1xRTT signal strength on a CDMA device falls below a specified threshold for a specified amount of time. Optionally, a subsequent alarm is triggered when signal strength returns to the optimal state above the threshold.

rssi_1xevdo

Triggers alarms when the average RSSI 1xEVDO signal strength on a CDMA device falls below a specified threshold for a specified amount of time. Optionally, a subsequent alarm is triggered when signal strength returns to the optimal state above the threshold.

ecio_1xrtt

Triggers alarms when the average Ec/Io 1xRTT signal strength on a CDMA device falls below a specified threshold for a specified amount of time. Optionally, a subsequent alarm is triggered when signal strength returns to the optimal state above the threshold.

ecio_1xevdo

Triggers alarms when the average Ec/Io 1xEVDO signal strength on a CDMA device falls below a specified threshold for a specified amount of time. Optionally, a subsequent alarm is triggered when signal strength returns to the optimal state above the threshold.

cell_data

Triggers alarms based on cellular data exchanged in an amount of time.

mobile_temp

Triggers alarms when the mobile temperature goes above a specified threshold. Optionally, a subsequent alarm is triggered when signal strength returns to a temperature below the threshold.

1xevdo_unavail

Triggers alarms when 1xEVDO service is unavailable for a specified amount of time. Optionally, a subsequent alarm is triggered when service becomes available again.

mobile_unavail

Triggers alarms when mobile service is unavailable (not registered on the home network) for a specified amount of time. Optionally, a subsequent alarm is triggered

when service becomes available again.

mobile_cfg_chng

Triggers alarms when the mobile configuration is changed.

GPIO pin state-based alarm options

In GPIO mode, alarms are triggered when there are transitions between states for GPIO pins. These options allow you set which GPIO pins' transitions trigger alarms. Setting alarms in GPIO mode is supported in some but not all Digi devices.

pins=*list of pins*

A list of GPIO pins that trigger alarms.

highpins=*list of highpins*

A list of GPIO pins that trigger alarms when a pin's signal is high.

lowpins=*list of lowpins*

A list of GPIO pins that trigger alarms when a pin's signal is low.

pin{*n*}={{high|1}|{low|0}|{ignore|x}}

An alternative way to specify the action of a given GPIO pin, where ***n*** is the pin number.

high or 1

The pin will trigger an alarm when the pin's signal is high.

low or 0

The pin will trigger an alarm when the pin's signal is low.

Ignore or x

The pin will not trigger an alarm.

The default is **ignore**.

reminder={on|off}

The type of reminder sent.

on

An email or SNMP trap is sent periodically while the alarm-triggering event is active. The interval is based on the value of the **reminder_interval** option.

off

An email or SNMP trap is sent only when an alarm is triggered.

reminder_interval=*seconds*

The minimum reminder interval in seconds. Indicates how often an email or SNMP trap is sent when the **reminder** option is set to **on** and an alarm-triggering event is active.

trigger_interval=*seconds*

The minimum trigger interval in seconds. If the **reminder** option is set to **off**, this option indicates the minimum amount of time that is allowed between alarm-triggered emails or SNMP traps.

Data pattern matching-based alarm options

In data pattern match mode, an alarm will be triggered when a pattern is found in the stream of serial data. These options are used for setting alarms in data pattern match mode:

mode=match

Sets the alarm to match mode.

match=*string*

A string that triggers an alarm if the data pattern is found in the incoming serial stream. The maximum length of this string is **40** characters, including escape sequences for special characters. For more details on the escape sequences, see "Entering Special Characters in String Values" on page 11. The maximum parsed length of this string is 10 characters. That is, this string must reduce down to a 10-character string when the escape sequences are processed.

port=*port_range*

Specifies the port or range of ports that will be monitored for the specified match string.

Signal-strength-based alarm options

Options for setting alarms based on signal strength are supported in cellular-enabled products only. Note that not all signal-strength options are available with all radios. To determine which signals can be monitored, use the “display mobile” command. If a signal is displayed, it can be monitored.

mode=rssi_gsm

Sets the alarm to signal-strength mode for a GSM radio.

mode=rssi_1xrtt

Sets the alarm to 1xRTT RSSI signal-strength mode for a CDMA radio.

mode=rssi_1xevdo

Sets the alarm to 1xEV-DO RSSI signal-strength mode for a CDMA radio.

mode=ecio_1xrtt

Sets the alarm to 1xRTT Ec/Io signal-strength mode for a CDMA radio.

mode=ecio_1xevdo

Sets the alarm to 1xEV-DO Ec/Io signal-strength mode for a CDMA radio.

signal_strength_threshold=*threshold*

The threshold average signal strength., This is measured in dBm for **rssi_gsm**, **rssi_1xrtt**, and **rssi_1xevdo**—typically -120 dBm to -40 dBm—and dB for **ecio_1xrtt** and **ecio_1xevdo**—typically -24 dB to -2 dB.

Note that a value of **0 dB** reported by a **display mobile** command when there is no signal strength should properly be interpreted as the minimum value, which is **-120 dBm** or **-24 dB**.

time=*time*

The time in minutes that the average signal strength stays below the threshold specified on the **signal_strength_threshold** option.

optimal_alarms_enabled={on|off}

If optimal alarms are enabled, an optimal alarm will also be sent when the signal strength returns to a value that is above the specified threshold. Default is **off**.

Cellular data traffic-based alarm options

These options for setting alarms based on cellular data traffic are supported in Digi Cellular Family products only.

mode=cell_data

Sets the alarm to cellular-data mode.

cell_data=*byte count threshold*

The number of bytes of cellular data to be counted before triggering an alarm.

cell_data_type={receive|transmit|total}

The type of cellular data to be counted.

receive

Data received by the Digi device.

transmit

Data transmitted by the Digi device.

total

The total data received and transmitted the Digi device.

time=*max time*

The time, in minutes, during which bytes of cellular data are counted.

Mobile temperature-based alarm options

These options for setting alarms based on the radio's reported temperature are supported in Digi Cellular Family products only. Note that temperature-based options are not available with all radios. To determine if your radio supports these alarms, use the **display mobile** command. If the temperature is displayed, it can be monitored.

mode=mobile_temp

Sets the alarm to mobile temperature mode.

temperature=*temperature*

The temperature threshold in degrees Celsius.

optimal_alarms_enabled={on|off}

If optimal alarms are enabled, an optimal alarm will also be sent when the temperature returns to a value that is below the specified threshold. Default is **off**.

Mobile service-based alarm options

These options for setting alarms based on the service state of the radio are supported in Digi Cellular Family products only. Note that some service-based options are not available with all radios. To determine if your radio supports these alarms, use the **display mobile** command. If the service is displayed, it can be monitored.

mode=1xevdo_unavail

Sets the alarm to monitor 1xEV-DO service.

mode=mobile_unavail

Sets the alarm to monitor registration status. Any registration status other than **Registered (Home Network)** is considered to be a mobile unavailable status.

time=*time*

The time in minutes that the specified service has been unavailable.

optimal_alarms_enabled={on|off}

If optimal alarms are enabled, an optimal alarm will also be sent when the service being monitored is reestablished. Default is **off**.

Mobile configuration change alarm options

This option for setting alarms based on a change to any of the mobile-oriented configurations is supported in the cellular-enabled products only. The following configuration items are monitored: PPP settings associated with the mobile radio (**set mobileppp** and **set ppp**); PPP Bridge settings (**set passthrough**); Mobile settings (**set surelink**); or mobile provisioning, manual or IOTA.

mode=mobile_cfg_chng

Sets the alarm to monitor mobile configuration changes.

Examples

Set a GPIO alarm and send an email message or SNMP trap

This example shows how to set up a GPIO alarm to trigger when two GPIO pins go high, and sending an email message when they do. It also shows how to change from sending an email message when the alarm condition occurs to issuing an SNMP trap.

1. Turn off alarms and set global email properties (this is done to avoid false error conditions triggering alarms):

```
#> set alarm state=off mailserverip=10.0.0.1 from=myemail@digi.com
```

2. Set alarm 1 mode to GPIO mode:

```
#> set alarm range=1 mode=gpio
```

3. Set alarm 1 to designate which pins trigger alarm:

```
#> set alarm range=1 pin2=high pin3=1
```

```
#> set alarm range=1 highpins=2,3
```

4. Set alarm 1 to send an email message when the alarm condition is met, and enable alarm 1:

```
#> set alarm range=1 active=on type=email to=destination@digi.com
subject=”Alarm 1 triggered”
```

5. Change alarm 1 to send an SNMP trap:

```
#> set alarm range=1 highpins=2,3 type=snmptrap
```

6. Enable alarms:

```
#> set alarm state=on
```

Set up signal-strength and cellular-traffic alarms and send them to Device Cloud

This example shows how to set up two alarm and have them sent to Device Cloud. It configures two alarms:
Alarm 1 for is based on a threshold signal strength value (**rssi**), and
Alarm 2 is based on cellular data traffic (**cell_data**).

1. Disable alarms during configuration:

```
#> set alarm state=off
```

2. Set **alarm 1** to trigger when average GSM rssi drops below **-80** dB for at least **20** minutes. Turn **alarm 1** on.

```
#> set alarm range=1 active=on mode=rssi_gsm sig_strength_threshold=-
80 time=20
```

3. Set **alarm 2** to trigger when more than **10000** bytes are sent in a period of **5** minutes. Turn **alarm 2** on.

```
#> set alarm range=2 active=on mode=cell_data cell_data=10000 cell_
data_type=transmit time=5
```

4. Set **alarm 3** to trigger when mobile radio temperature exceeds the default thresholds. Turn **alarm 3** on.

```
#> set alarm range=3 active=on mode=mobile_temp
```

5. Set **alarm 4** to trigger when 1xEV-DO service is unavailable for a period of **3** minutes and also trigger when service becomes available again. Turn **alarm 4** on.

```
#> set alarm range=4 active=on mode=1xevdo_unavail time=3 optimal_
alarms_enabled=yes
```

6. Set **alarm 5** to trigger when the mobile configuration has been changed. Turn **alarm 5** on.

```
#> set alarm range=5 active=on mode=mobile_config_change
```

7. Set all alarms to be sent to Device Cloud, and turn on alarms:

```
#> set alarm state=on dcloud=on
```

See also

- revert: The **revert alarm** command reverts the settings configured by this command.
- set gpio: This command determines whether pins act as GPIO input, GPIO output, or standard serial.
- set smscell
- set snmp
- show: The **show alarm** command shows the current alarm settings in the Digi device.

set autoconnect

Purpose

Used to establish an automatic connection (autoconnection) between the serial port and a remote network destination, and to display current autoconnect settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display autoconnect settings for the line on which they are logged in: **set permissions s-autoconnect=r-self**.
- For a user to display autoconnect settings for any line: **set permissions s-autoconnect=read**.
- For a user to display and set the autoconnect settings for the line on which they are logged in: **set permissions s-autoconnect=rw-self**.
- For a user to display autoconnect settings for any line, and set the autoconnect settings for the line on which the user is logged in: **set permissions s-autoconnect=w-self-r**.
- For a user to display and set the autoconnect settings on any line: **set permissions s-autoconnect=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Configure autoconnect

```
set autoconnect [port=range]
  [state={on|off}]
  [trigger={always|destination|data|dcd|dsr|string}]
  [service={raw|rlogin|ssl|telnet}]
  [description={string}]
  [ipaddress=ipaddress]
  [ipport=ipport]
  [connect_on_string=string]
  [flush_string={on|off}]
  [keepalive={on|off}]
  [nodelay=on|off]
```

Display autoconnect settings

```
set autoconnect [port=range]
```

Options

port=*range*

The serial port to which the autoconnect settings apply. Optional on a single-port device.

state={on|off}

Enables or disables the autoconnect feature.

on

Enables the autoconnect feature.

off

Disables the autoconnect feature.

The default is off.

If you are using the serial port for another purpose, it is recommended this value be set to **off**.

trigger={always|destination|data|dcd|dsr|string}

Indicates which events from the serial port will trigger a network connection to occur.

always

The serial port will continually attempt to keep a connection to a remote network destination active.

destination

The serial port will attempt a network connection whenever data arrives from the network destination specified by the **ipaddress** option.

data

The serial port will attempt a network connection whenever data arrives on the serial port.

dcd

The serial port will attempt a network connection whenever the serial port's DCD signal is **asserted**.

dsr

The serial port will attempt a network connection whenever the serial port's DSR signal is **asserted** or **raised**.

string

A connection will be made upon detecting a particular sting, specified by the **connect_on_string** option, in the data from the serial port.

The default is **always**.

service={raw|rlogin|ssl|telnet}

The type of network connection that will be established.

raw

A connection without any special processing occurs.

rlogin

A remote login (rlogin) connection occurs.

ssl

A secure connection conforming to SSL (Secure Sockets Layer) Version 3 and Transport Layer Security (TLS) Version 1 occurs.

telnet

A connection with Telnet processing occurs.

The default is **raw**.

description=*string*

A name for descriptive purposes only.

ipaddress=*ipaddress*

The IP address of the network destination to which a connection will be made.

ipport=ipport

The TCP port of the network destination to which a connection will be made.

connect_on_string=*string*

When the value of the **trigger** option is string, this option specifies the string that must be found in the serial data in order for a connection to occur. The maximum length of this string is **32** characters, including escape sequences for special characters. For more details on the escape sequences, see "Entering Special Characters in String Values" on page 11. The maximum parsed length of this string is 32 characters. That is, this string must reduce down to a 32-character string when the escape sequences are processed.

flush_string={on|off}

Indicates whether the connect string, specified by the **connect_on_string** option, is flushed or sent over the newly established connection.

on

The connect string is flushed.

off

The connect string is sent over the newly established connection.

The default is on.

keepalive={on|off}

Indicates whether or not TCP keepalives will be sent for the specified range of clients. If set to on, keepalives will be sent, if it is off, keepalives will not be sent.

Configurable TCP keepalive parameters, for example, how many keepalives to send and when to send them are configured globally via the **set network** command (see "set network" on page 288).

nodelay={on|off}

Used to allow unacknowledged or smaller than maximum segment sized data to be sent.

Note The **nodelay** option disables Nagle's algorithm, which is on by default, for some TCP services. The purpose of Nagle's algorithm is to reduce the number of small packets sent. Nagle's algorithm says to hold on to outgoing data when there is either unacknowledged sent data or there is less than maximum segment size (typically around 1500 bytes for Ethernet) worth of data to be sent. While Nagle's algorithm does a good job at keeping transmission efficient, but there are times where it is desirable to disable it.

Examples

Set autoconnect on with trigger

This example shows setting autoconnect to connect to the TCP port (**2101**) of the network IP destination when data arrives on the serial port.

```
#> set autoconnect state=on trigger=data ipaddress=10.0.0.1 ipport=2101
```

Allow outgoing data that is either unacknowledged or less than maximum segment size

```
#> set autoconnect port=1 nodelay=on
```

See also

- revert: The **revert autoconnect** command reverts the settings configured by this command.
- set network
- set serial
- set tcpserial
- show: The **show autoconnect** command shows the current autoconnect settings in a Digi device.

set buffer

Purpose

Configures buffering settings on a port or displays the port buffer configuration settings on all ports and the total memory available for buffers. The port buffering feature allows you to monitor incoming ASCII serial data in log form.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the port buffering settings for the line on which they are logged in: **set permissions buffers=r-self**
- For a user to display the port buffering settings for any line: **set permissions buffers=read**
- For a user to display and set the port buffering settings for the line on which they are logged in: **set permissions buffers=rw-self**
- For a user to display the port buffering settings for any line, and set port buffering settings for the line on which the user is logged in: **set permissions buffers=w-self-r**
- For a user to display and set the port buffering settings on any line: **set permissions buffers=rw**

See set permissions for details on setting user permissions for commands.

Syntax

Configure port buffering

```
set buffer [clear] [port=range] [size=size in kilobytes]
  [state={on|off|paused}]
```

Display port buffering settings

```
set buffer [port=port]
```

Options

clear

Clears the contents of the specified buffer.

port=*range*

The port or ports to which the command applies.

size=*size in kilobytes*

The size in kilobytes to configure the buffer. Settings are configurable in 2-kilobyte increments. The maximum size is **64** kilobytes. The default is **32** kilobytes.

state={on|off|paused}

The buffering state, which can be any of the following:

on

The data is buffered.

off

The data is not buffered and all data is cleared from the buffer.

paused

The data is not buffered, but data in the buffer is not cleared.

Examples

Display port buffer configuration for all ports

```
#> set buffer
tty     state  size(kb)   % usage

  1       off        32         0
```

Configure buffers

In this example, the set buffer command sets the buffer state for port **1** to on mode and the buffer size to **64** kilobytes.

```
#> set buffer port=1 state=on size=64
```

See also

- display buffers
- revert: The **revert buffer** command reverts the settings configured by this command.
- show: The **show buffer** command shows the current buffering settings in a Digi device.

set camera

Purpose

Configures the camera settings for a Watchport® camera connected to a Digi device.

Digi devices support connections to one Digi Watchport V2 or V3 USB camera.

When the camera is connected to the Digi device and configured, the current image from the camera is available directly from the device at: http://device-ip/FS/dev/camera/0

Video from the camera is available by streaming the camera data to a TCP server application or from a Java applet by clicking on the video link in the config page. See below.

Note Access to the camera is not subject to user authentication. Turning on security for the web interface does not limit direct access to the camera.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-camera=read** to display camera settings, and to **set permissions s-camera=rw** to display and configure camera settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure basic options

```
set camera [state={on|off}]
  [resolution={128x96|160x120|176x144|320x240|
  352x288|640x360|640x480}]
  [frame_delay=0-65535 milliseconds]
  [quality=0-100]
  [frequency={50|60}]
  [tcp_client_state={on|off}]
  [host=host name or ip address of tcp server]
  [port=tcp server port to connect to]
```

Configure advanced options

```
set camera [agc={off|average|center|peak}]
  [white_balance={manual|auto|daylight|incandescent|fluorescent}]
  [backlight=0-255]
  [brightness=0-255]
  [gamma=0-255]
  [saturation=0-255]
  [sharpness=0-255]
  [contrast=0-255]
  [decimate={on|off}]
```

Display current camera settings

```
set camera
```

Options

Basic options

state={on|off}

Turns the camera on or off. When disabled, all camera activity stops and all memory used will be freed.

resolution={128x96 |160x120|176x144|320x240 352x288|640x360|640x480}

The image resolution.

frame_delay=0-65535 milliseconds

Configures how often a new image is retrieved from the camera.

The minimum time between frames (in milliseconds). The actual delay time between frames will be this number or greater. The camera will automatically increase this value as needed, such as in low-light conditions.

This delay time is the inverse of frames per second. For instance, if you wish to set the camera to process at a maximum of 5 frames per second, set the frame delay to **200** (1/5 = 0.2 second = 200 ms).

quality=0-100

Indicates the image quality of the camera image. **100** is the highest image quality; **0** is the worst image quality. The higher the quality, the more memory and other device resources will be required by the camera. It is recommended to pick a quality between **20** and **80**. Quality above 80 results in much larger images than lower qualities, which in turn results in lower overall performance and increased memory use.

frequency={50|60}

This option is used to reduce flicker from fluorescent lighting only. Set this field to match the frequency of the power supply connected to the Digi device In North America, use **60**. In Europe, use **50**.

tcp_client_state={on|off}

Enables or disables the camera sending its images to a remote server via TCP. The TCP server application must conform to the protocol sent by this device. The protocol is: on connect, the TCP client sends a protocol id of four bytes: **0x85ce4a71**, followed by a protocol version of 4 bytes: **0x00000010**. After this, images are sent over and over in the form of 4 bytes containing the length of the JPEG image to follow, followed by the JPEG image. An example Java application that reads images from a device is available at the Digi website, www.digi.com.

host=*host name* or *ip address of tcp server*

The name of the remote TCP Server to receive image data.

port=*tcp server port to connect to*

The network port number for the remote TCP server that receives image data. The default port is **22222**.

Advanced options

Advanced camera settings; Digi recommends leaving these camera settings at their defaults. They can be modified for specific needs by advanced users but do not need to be modified by most users.

agc={off|average|center|peak}

Automatic gain control.

white_balance={manual|auto|daylight|incandescent|fluorescent}

White balance.

backlight=0-255
Backlighting control.
brightness=0-255
Brightness level.
gamma=0-255
Gamma level.
saturation=0-255
Saturation level.
sharpness=0-255
Sharpness level.
contrast=0-255
Contrast level.
decimate={on|off}
Decimation control.

Examples

Configure camera images to a resolution of 640x480

```
#> set camera resolution=640x480
```

Configure camera to image quality of 50

```
#> set camhera quality=50
```

Configure camera to send all of its images to the server named "camera-server" on port 22222

```
#> set camera tcp_client_state=on host=camera-server port=22222
```

See also

- info camera: The **info camera** command shows information about any attached camera.
- revert: The **revert camera** command reverts the settings configured by this command.
- show: The **show accesscontrol** command shows current camera settings in a Digi device.

set clocksource

Purpose

Configures access to various external time sources that can be used to set and maintain time on the device.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-time-source=read** to display time source settings, and to **set permissions s-time-source=rw** to display and configure time source settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure a time source entry

```
set clocksource [range={1|2|3|4|5}]
  [type={unconfigured|cellular|sntp}]
  [state={on|off}]
  [interval=interval]
  [fqdn=fqdn]
```

Display a time source entry

```
set clocksource [range={1|2|3|4|5}]
```

Options

range={1|2|3|4|5}

Specifies the time source entry being created or modified.

type={unconfigured|cellular|sntp}

The type of clock source.

unconfigured

No clock source.

cellular

Use the cellular modem as a clock source.

sntp

Use an NTP/SNTP server as a clock source.

state={on|off}

Enable or disables a clock source entry.

on

This clock source entry is enabled.

off

This clock source entry is disabled.

interval=*interval*

The rate in seconds at which the time source will be polled for time. If multiple clock sources are enabled, an entry with a shorter poll time will have more influence on the device's time than an entry with a longer poll time

fqdn=*fqdn*

Specifies the Fully-Qualified Domain Name or IP address of the NTP/SNTP server.

Examples

```
# > set clocksource range=3 type=cellular state=on interval=3600
```

See also

- info time
- revert: The **revert clocksource** command reverts the settings configured by this command.
- set time
- set timemgmt
- show: The **show clocksource** command shows the current time-source settings in a Digi device.

set dcloud_msgservice

Purpose

Configures the ability to send and receive machine-to-machine (M2M) messages between Device Cloud (formerly, and the Device Cloud-registered device, known as Device Cloud SMS support. These settings are used in sending Short Message Service (SMS) messages to Device Cloud.

The **iridium** settings configure power management and message handling for the Iridium satellite modem in a Digi device.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions-dcloud-sms=read** to display the Device Cloud message service settings, and **set permissions s-dcloud-sms=rw** to display and configure Device Cloud message service settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set dcloud_msgservice [sms_state={on|off}]
  [sms_restrict_sender={on|off}]
  [sms_phnum=string]
  [sms_service_identifier=string]
  [iridium_state={on|off}]
  [iridium_automatically_manage_power={on|off}]
  [iridium_poll_rate=integer]
  [iridium_set_power={on|off}]
```

Options

sms_state={on|off}]

Enables or disables Device Cloud SMS support.

sms_restrict_sender={on|off}

If set to **on**, the Digi device will only process inbound messages for Device Cloud if they are from the phone number specified on the **sms_phnum** option. Messages from other phone numbers will be passed on to other SMS services on the Digi device.

sms_phnum=*string*

The phone number or short code of the Device Cloud server. For Device Cloud, the short code is **64225**.

sms_service_identifier=*string*

The service identifier (prefix) of the Device Cloud server.

This field is an optional setting and is used in cases where there is a shared short code in use, and an identifier (prefix) is required to redirect a message to a specific service under that short code. The default value is **idgp**. Use of the service identifier **idgp** is mandatory when the **sms_phnum** value is **64225**.

iridium_state={on|off

Enables or disables Device Cloud Iridium support.

iridium_automatically_manage_power={on|off}

Enables or disables having the Digi device automatically manage the power to the Iridium modem. This means power will only be turned on prior to sending a message, and turned off once the message has been sent. If this is NOT selected, the power will be turned on or off based on the setting for the **iridium power={on|off}** command for the device**.** If Device Cloud Iridium support is NOT enabled, the power will be turned off when the device starts up. If **iridium_automatically_manage_power** is turned on, the device will only be able to receive messages when polling, or if a message is sent. (that is, the modem will be off at all other times, so RING alerts will in general be missed). Turning on and off the modem can take up to a minute, which may result in delays sending messages.

iridium_poll_rate=*integer*

The number of minutes between poll attempts to the Iridium network. An entry of 0 disables polling. Polling may be needed to check for pending receive messages if the **RING** alert is missed. A **RING** alert is a notification from the network that a Mobile Terminated SBD Message is queued at the Gateway. Polling is chargeable. Every poll is the same as sending a message in terms of Iridium costs.

iridium_set_power={on|off}

Enables or disables having the Digi device set power to the Iridium modem. If the Digi device is configured to automatically manage power by setting **iridium_automatically_manage_power** to **on**, this option has no effect.

Example

```
#> set dcloud_msgservice sms_state=on sms_restrict_sender=on sms_phnum=64225 sms_
service_identifier=idgp
```

See also

- revert: The **revert dcloud_msgservice** command reverts the settings configured by this command.
- iridium for the **iridium power={on|off}** command.
- show: The **show dcloud_msgservice** command shows the current Device Cloud SMS settings in a Digi device.
- The Device Cloud SMS section in the *Device Cloud User Guide*.
- The Device Cloud SMS section in the *Device Cloud Programming Guide*.

set ddns

Purpose

A Dynamic DNS ((DDNS) service allows a user whose IP address is dynamically assigned to be located by a host or domain name. Before a DDNS service may be used, and before using this command, you must create an account with the DDNS service provider. The provider gives you account information such as username and password. Use this account information to register your IP address and update it as it changes.

A DDNS service provider may support registration of both public and private IP addresses. If the service provider does not support private IP address registration, and you attempt to register a private IP address (for example, **192.168.x.x** or **10.x.x.x**), the service provider may reject your update requests. If the service provider permits registration of private IP addresses, and you register a private IP address, your Digi device may be accessible (by resolving the associated hostname) only from other hosts with access to that private IP subnetwork.

Your Digi device monitors the IP address it is assigned for a selected network interface. It typically updates the DDNS service or server automatically, but only when its IP address has changed from the IP address is previously registered with that service.

DDNS service providers may consider frequent updates to be an abuse of their service. In such a circumstance, the service provider may block updates from the abusive host for some period of time, or until the customer contacts the provider. Observe the requirements of the DDNS service provider to ensure compliance with possible abuse guidelines.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ddnsupdater=read** to display Dynamic DNS settings, and to **set permissions s-ddnsupdater=rw**. to display and configure Dynamic DNS settings. See set permissions for details on setting user permissions for commands.

Syntax

Set basic DDNS settings

```
set ddns [service={disabled|dyndnsorg}]
  [ifname=network interface name]
  [action={updatenow|clearstatus}]
```

Note If **action** is specified, other options are not used.

Set service settings for Dynamic DNS (service=dyndnsorg):

```
set ddns [ddconntype={standardhttp|alternatehttp|securehttp}]
  [ddsystem={dyndns|statdns|custom}]
  [ddusername=user name]
  [ddpassword=password for DynDNS.org account]
  [ddhostname=full host name]
  [ddwildcard={off|on|nochg}]
```

Display current DDNS settings

```
set ddns
```

Options

Basic DDNS settings

service={disabled|dyndnsorg}

Specifies the DDNS service used for handling dynamic DNS updates, or disables use of the DDNS service.

disabled

Turns off the Dynamic DDNS service. This the default setting for the Dynamic DNS feature. Use this option if you have configured the Dynamic DNS feature and you want to temporarily turn off the service for some reason.

dyndnsorg

Update the DDNS service at **DynDNS.org**. When you select a specific DDNS service provider, you must also provide the related account information for that service provider.

ifname=*network interface name*

The network interface name; allowed network interfaces vary among products.

action={updatenow|clearstatus}

The action to be performed on the DDNS service. If you specify**action**, other command options are not used.

updatenow

Force a DDNS service update now.

clearstatus

Clears the last information returned from the Dynamic DNS service from your Digi device.

DynDNS.org service settings

These settings are specific to your account information with **DynDNS.org**; please consult their website for more information on account terms and settings.

ddconntype={standardhttp|alternatehttp|securehttp}

The method to use to connect to the **DynDNS.org** server:

standardhttp

Connect to the Standard HTTP port (**80**).

alternatehttp

Connect to the Alternate HTTP port (**8245**).

securehttp

Connect to the Secure HTTPS port (**443**).

ddsystem={dyndns|statdns|custom}

The **DynDNS.org** system to use for the update.

dyndns

Update a Dynamic DNS host name.

statdns

Update a Static DNS host name.

custom

Update a Custom DNS host name.

ddusername=*user name*

The user name for the **DynDNS.org** account.

ddpassword=password

The password for **DynDNS.org** account.

ddhostname=full host name

The full host name to update for **DynDNS.org** account, for example, **myhost.dyndns.net**.

ddwildcard={off|on|nochg}

Enables/disables wildcards for this host. The wildcard aliases ***.yourhost.ourdomain.tld** to the same address as **yourhost.ourdomain.tld**. Using this option has the same effect as selecting the wildcard option on the **DynDNS.org** web site. To leave the wildcard option unchanged from the current selection on their website, use the “no change” option (the **nochg** keyword) in the device settings. **DynDNS.org** support for this option may vary according to the DynDNS system you are registered to use.

off

Disables wildcards.

on

Enables wildcards.

nochg

Specifies that there should be no change to service setting from the current selection for wildcards in the DDNS settings at the **DynDNS.org** Web site.

Examples

This example shows **set ddns** being used to display the current status of the Dynamic DNS service:

```
#> set ddns

 DDNS Service Update Configuration :

    service     : dyndnsorg
    ifname      : mobile0

    ddconntype  : securehttp
```

```
    ddsystem    : statdns
    ddusername  : "test"
    ddpassword  : (Not shown for security reasons)
    ddhostname  : "test-static.dnsalias.com,test-static.dnsalias.org"
    ddwildcard  : nochg

 Current IP address: 166.213.228.220 (ppp1)
```

```
 Most recent DDNS service update status:

    Service              : DynDNS.org
    IP address reported  : 166.213.228.220
    Update status        : successful
    Result information   : [good] The update was successful.
    Raw result data      :
      good 166.213.228.220
      good 166.213.228.220
```

```
 Most recent DDNS service update log message:
    IP address for "ppp1" is now 166.213.228.220, but no DDNS update is needed
(last reported IP address is unchanged).
```

See also

- display ddns
- revert: The **revert ddns** command reverts the settings configured by this command.
- show: The **show ddns** command shows current Dynamic DNS settings in a Digi device.
- Dynamic DNS resources available from your service provider, such as glossary definitions, FAQs, knowledge base articles, and tips for managing your account.

set devicesecurity

Purpose

Used to set or display Device Cloud device security settings. These settings involve the security-related features of cryptographic identity verification, or authentication, and encryption of the protocol used to manage and pass data between devices and the Device Cloud.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-devicesecurity=read** to display device security settings, and **set permissions s-devicesecurity=rw** to display and configure device security settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Device Cloud device security settings

```
set devicesecurity [identityverificationform=[{simple|password}]
  [password=password]
```

Display current Device Cloud device security settings

```
set devicesecurity
```

Options

identityverificationform={simple|password}

Used to specify the type of authentication used by the server.

simple

The device sends its device ID to Device Cloud.

password

The device and the Device Cloud server shares a secret password, specified by the **password** option.

password=*password*

A password of up to **63** characters that is a shared secret password with the Device Cloud server.

Examples

Set device security settings

```
#> set devicesecurity identityverificationform=password password=oas952xeo
```

See also

- display dcloud
- revert: The **revert devicesecurity** command reverts the settings configured by this command.
- set mgmtconnection
- set mgmtglobal
- set mgmtnetwork
- show: The **show devicesecurity** command shows the current Device Cloud device security settings in a Digi device.
- The *Device Cloud User Guide.*
- The *Device Cloud Programming Guide.*

set dhcpserver

Purpose

Configures the DHCP server settings for the Digi device. A DHCP server allows other devices or hosts on the same local network as the Digi device to be assigned dynamic IP addresses. This DHCP server supports a single subnetwork scope. The Digi DHCP server scope can be on a LAN interface only. That is, the Dig DHCP server serves IP addresses to DHCP clients on the Ethernet side of the Digi device only. Supported interfaces include:

- **eth0**: Ethernet (primary/only Ethernet)
- **eth1**: Ethernet (second Ethernet, such as on Connect ES)
- **wln0**: Wi-Fi

The Digi DHCP server and this command do not serve WAN or point-to-point interfaces, such as **pppX** (PPP interface (serial)), **mobile0** (mobile/cellular), or **wmx0** (WiMAX).

The DHCP server operates only if the Digi device is configured to use static IP address configuration. For information on how to configure static IP settings, see set network and the help for the web interface's Network Configuration settings.

Once configured, the DHCP server is managed through the **dhcpserver** command. See dhcpserver.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-dhcpserver=read** to display DHCP server settings, and **set permissions s-dhcpserver=rw** to display and configure DHCP server settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure the DHCP server: basic settings

```
set dhcpserver item={scope|reservation|exclusion}
  [action={set|revert}]
```

item={scope|reservation|exclusion}

Specifies the configuration settings to which the command will be applied. The **item** option defaults to **scope** if not specified.

action={set|revert}

Specifies the action to be performed by the **set dhcpserver** command. This option defaults to **set** if not specified.

action={set|revert}

Specifies the action to be performed by the **set dhcpserver** command.

set

Sets the DHCP configuration settings to the specified values. This is the default setting that is used if this option is not specified.

revert

Resets the configuration settings for the specified item to default values.

Configure the DHCP server scope ("item=scope")

To enable a scope, or a reservation or exclusion entry within a scope, all of its parameters must be specified and valid. A scope or entry can be enabled only if it is completely valid.

```
set dhcpserver item=scope
  [action={set|revert}]
  [name=scope interface name]
  [enabled={on|off}]
  [startip=ip address]
  [endip=ip address]
  [leasetime={time|infinite}] (in seconds, 0=default (86400)}
  [offerdelay=0-5000] (time in milliseconds, default 500}
  [conflictdetect={on|off}]
  [dnsproxy={on|off}]
  [gatewayopt={ifaddress|ifgateway|gwoptip|none}]
  [gwoptip=ip address]
```

Configure the scope's address reservations ("item=reservation")

In this type of configuration, the DHCP server assigns a particular IP address to a Digi device, rather than a random address from a pool.

```
set dhcpserver item=reservation
  [action={set|revert}]
  [range={1-16|all}]
  [enabled={on|off}]
  [ip=ip address]
  [clientid=client MAC address] {e.g., 00:40:9D:12:34:56}
  [leasetime={time|infinite}] {in seconds, 0=default (use scope's time)}
```

Configure the scope's address exclusions ("item=exclusion")

```
set dhcpserver item=exclusion
  [action={set|revert}]
  [range={1-4|all}]
  [enabled={on|off}]
  [startip=ip address]
  [endip=ip address]
```

Option rules on scope, reservation, or exclusion entries

For a reservation or exclusion entry, the **range** option selects the specific entry to which the action is to be applied.

To enable a scope, or a reservation or exclusion entry within a scope, all of its parameters must be specified and valid. A scope or entry can be enabled only if it is completely valid.

Display current DHCP server settings

```
set dhcpserver
```

Options

Options for configuring the DHCP server scope ("item=scope")

item=scope

Specifies that the DHCP server configuration settings apply to the scope of IP addresses for a network. A scope is the full consecutive range of possible IP addresses for a network. A scope typically defines a single physical subnet on your network, to which DHCP services are offered. A scope is the primary way for the DHCP server to manage distribution and assignment of IP addresses and related configuration parameters to its clients on the network.

action={set|revert}

Specifies the action to be performed by the **set dhcpserver** command.

set

Sets the DHCP configuration settings to the specified values. This is the default setting that is used if this option is not specified.

revert

Resets the configuration settings for the specified item to default values.

name=*scope interface name*

A valid interface name for the scope. The default is **eth0**.

enabled={on|off}

Enables the DHCP server feature on this Digi device. Note that for the DHCP server to operate, the Digi device must be configured to use a static IP address.

startip=*ip address*

The first IP address in the pool.

endip=*ip address*

The last address in the pool. The addresses in the range specified by **startip** and **endip** must be in the same subnet as the Digi device.

leasetime={time|infinite}

The length, in seconds, of the leases for the scope being served by this DHCP server. Specifying a time of **0** means that the default of **86400** seconds (**24** hours) will be used. A DHCP client may request a lease duration other than this setting, and the DHCP server will grant that request, if possible.

offerdelay=0-5000

The interval of time, in milliseconds, to delay before offering a lease to a new client. The range for this delay is **0** to **5000** milliseconds, and the default delay is **500** milliseconds. Use of this delay permits the Digi device to reside on a network with other DHCP servers, yet not offer leases to new clients unless the other DHCP servers do not make such an offer. This provides a measure of protection against inadvertently connecting a Digi device to a network that is running its own DHCP server, and thereby offering leases to clients in a manner inconsistent with that network.

conflictdetect={on|off}

When a DHCP client requests a new IP address lease, before offering an IP address to that client, use **ping** to test whether that IP address is already in use by another host on the network but is unknown to the DHCP Server. If an IP address is determined to be in use, it is marked as **Unavailable** for a period of time, and it will not be offered to any client while in this state.

Enabling this test adds approximately one second of delay before the IP address is offered to the client, since the **ping** test must not receive a valid reply for that test to successfully determine that the IP address is not already in use.

This option is **off** (disabled) by default. This option does not apply to Static Lease Reservations, since the **ping** test is not used for them.

dnsproxy={on|off}

Enables or disables sending the DHCP server IP address as DNS proxy. Default is **on**.

Note The DHCP server IP address is not sent if DNS proxy is disabled. Basically, this means that if the DNS Proxy feature is disabled by the **set dnsproxy** command, then the DHCP server will not include its own IP address as a DNS server in the lease it sends to its DHCP clients.

---

gatewayopt={ifaddress|ifgateway|gwoptip|none}

Gateway option. The purpose of the **gatewayopt** and **gwoptip** options is to enhance the DHCP server's configurability in terms of the default gateway that the server will provide to its DHCP clients for DHCP Option 3, Routers On Subnet. This option configures the DHCP Server to send DHCP Option 3: Routers on Subnet, to its DHCP clients in the lease information. Choices for this option include:

ifaddress

The IP address of the scope's network interface is sent to the client via DHCP Option 3. This choice is on (enabled) as the default.

ifgateway

The configured default gateway IP address for the scope's network interface is sent to the client via DHCP Option 3.

Send the configured IP address of the default gateway for the scope interface.

gwoptip

The gateway IP address specified by the **gwoptip** option is sent to the client via DHCP Option 3. This address must be reachable by the client or IP routing will not succeed.

none

No default gateway is sent to the client via DHCP Option 3.

gwoptip=*ip address*

If **gatewayopt=gwoptip** is specified, this option specifies the IP address of the gateway.

Options for configuring the scope's address reservations (“item=reservation”)

item=reservation

Specifies that the DHCP server configuration settings apply to the scope’s address reservations. You may use a reservation to create a permanent address lease assignment by the DHCP server. Reservations assure that a specified hardware device on the subnet can always use the same IP address. Address lease reservations associate a specific IP address with a specific client’s Ethernet MAC address.

action={set|revert}

Specifies the action to be performed by the **set dhcpserver** command.

set

Sets the DHCP configuration settings to the specified values. This is the default setting

that is used if this option is not specified.

revert

Resets the configuration settings for the specified item to default values. This effectively removes the entry specified by the **range** option.

range={1-16|all}

Selects the specific entry to which the action is to be applied.

enabled={on|off}

Enables the DHCP server feature on this Digi device. For the DHCP server to operate, the Digi device must be configured to use a static IP address.

ip=*ip address*

The IP address reserved for the client. This value must not be the same as the IP address of the DHCP Server itself.

clientid=*client MAC address*

The MAC address for the client, for example, **00:40:9D:12:34:56**.

leasetime={time|infinite}

The length, in seconds, of the leases for the scope being served by this DHCP server. Specifying a time of **0** means that the default of using the scope's lease time will be used. A DHCP client may request a lease duration other than this setting, and the DHCP server will grant that request, if possible. Leaving this option blank causes the default value specified in the scope to be used. That default changes whenever the scope changes.

Options for configuring the scope's address exclusions ("item=exclusion")

item=exclusion

Specifies that the DHCP server configuration settings apply to the scope's address exclusions. An exclusion range is a limited sequence of IP addresses within a scope, excluded from DHCP service offerings. Exclusion ranges assure that any addresses in these ranges are not offered by the server to DHCP clients on the network. Note that the IP address of the DHCP server itself will not be given out to any DHCP clients, even if it is within the range specified on this command.

action={set|revert}

Specifies the action to be performed by the **set dhcpserver** command.

set

Sets the DHCP configuration settings to the specified values. This is the default setting that is used if this option is not specified.

revert

Resets the configuration settings for the specified item to default values. This effectively removes the entry specified by the **range** option.

range={1-4|all}

Selects the specific entry to which the action is to be applied.

enabled={on|off}

Enables the DHCP server feature on this Digi device. Note that for the DHCP server to operate, the Digi device must be configured to use a static IP address.

startip=*ip address*

The first address in the exclusion block.

endip=*ip address*

The last address in the exclusion block. An IP address exclusion range must specify valid IP addresses in the subnetwork of the DHCP server. The DHCP server will ignore an exclusion that does not meet this requirement.

Examples

Configure the IP address range for the DHCP server scope and enable the DHCP server

```
set dhcpserver item=scope action=set enabled=on startip=10.30.1.150
endip=10.30.1.199
```

Since the **leasetime** and **offerdelay** options are not specified, the default values for them are used, unless they were previously changed to another value by use of a **set dhcpserver item=scope** command (or using the web UI).

Add an IP address reservation for a client

```
set dhcpserver item=reservation action=set range=1 enabled=on ip=10.30.1.195
clientid=00:09:26:19:51:05
```

Since the leasetime option is not specified, the DHCP server's scope lease time is used, unless a lease time was previously changed to a value by use of a **set dhcpserver item=reservation action=set range=1** command (or using the web UI).

Disable all reservations that were previously added:

set dhcpserver item=reservation action=set range=all enabled=off

Permanently remove a reservation that was previously added

```
set dhcpserver item=reservation action=revert range=1
```

Any client that has the lease for the reserved IP address that is removed in this manner, still keeps its lease. However, without the reservation, future address leases to that client are not guaranteed to be for this same IP address, which is the purpose of a reservation.

Add an IP address exclusion range for the scope:

```
set dhcpserver item=exclusion action=set range=1 enabled=on startip=10.30.1.170
endip=10.30.1.179
```

This exclusion instructs the DHCP server to not issue leases for the IP addresses from **10.30.1.170** to **10.30.1.179** inclusive. Note that reservation leases may be configured for any address in that range, and the DHCP server permits a lease of such an address to the correct client only. That is, reservations override exclusions.

Display current DHCP server settings

```
#> set dhcpserver

 DHCP Server Settings:
```

```
    server enabled       : on
    scope name           : eth0
    starting ip address  : 10.30.1.190
    ending ip address    : 10.30.1.198
    lease time           : 3600 (seconds)
    offer delay          : 500 (milliseconds)
    addr conflict detect : off
    send DNS proxy       : on
    gateway option       : ifaddress
    gateway opt ip addr  : 0.0.0.0 (N/A)
```

```
  Reservation Settings:

    idx enabled ip address      client id          lease time
    --- ------- --------------- -----------------  ----------
    1   on      10.30.1.135     00:40:9D:24:73:F8        3600
    2   on      10.30.1.195     00:09:26:19:51:05           0
    3   on      10.30.1.196     00:09:26:19:51:06           0
    4   on      10.30.1.197     00:09:26:19:51:07           0
    5   on      0.0.0.0         00:00:00:00:00:00           0
    6   on      0.0.0.0         00:00:00:00:00:00           0
    7   off     0.0.0.0         00:00:00:00:00:00           0
    8   off     0.0.0.0         00:00:00:00:00:00           0
    9   off     0.0.0.0         00:00:00:00:00:00           0
    10  off     0.0.0.0         00:00:00:00:00:00           0
    11  off     0.0.0.0         00:00:00:00:00:00           0
    12  off     0.0.0.0         00:00:00:00:00:00           0
    13  off     0.0.0.0         00:00:00:00:00:00           0
    14  off     0.0.0.0         00:00:00:00:00:00           0
    15  off     0.0.0.0         00:00:00:00:00:00           0
    16  off     0.0.0.0         00:00:00:00:00:00           0

    A reservation lease time of 0 means to use the scope's lease time.
```

```
Exclusion Settings:

    idx enabled start address   end address
    --- ------- --------------- ---------------
    1   off     0.0.0.0         0.0.0.0
    2   off     0.0.0.0         0.0.0.0
    3   off     0.0.0.0         0.0.0.0
    4   off     0.0.0.0         0.0.0.0
```

See also

- dhcpserver
- revert: The **revert dhcpserver** command reverts the settings configured by this command.
- set dnsproxy
- show: The **show dhcpserver** command shows the current DHCP server settings in a Digi device.

- In the web interface, the online help for Network settings; access from the main menu **Configuration > Network** and click the **Help** button/link in the upper right corner. The topic **Configuring DHCP Sever Settings** provides more information on DHCP terminology and managing DHCP server operation.

set dialserv

Purpose

Configures special behaviors on serial ports with DialServ devices attached.

Digi DialServ is an RJ-11 phone line simulator that allows legacy devices with built-in modems to communicate across LANs/WANs. This profile configures the Digi device to connect/tunnel serial data to an external host when the DialServ receives an incoming call, causes the DialServ to make outgoing calls, and tunnels TCP data from the incoming connection over the Dialserv when TCP traffic is received on the configured ports on the Digi device.

Important: Use of the **dialserv** serial port profile is **required** for DialServ interoperation. See "set profile" on page 324.

The connection wait time parameter specifies how much time the DialServ should wait for any outgoing connection to be made with the modem to which it is connected, after which it aborts the incoming TCP session that triggered the outgoing connection attempt.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-dialserv=read** to display DialServ settings, and **set permissions s-dialserv=rw** to display and configure DialServ settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set dialserv [ports=range]
  [init_script=initialization script]
  [connection_wait_time=time]
```

Options

ports=*range*

The port or ports to which the DialServ settings apply.

init_script=*initialization script*

The name of an initialization script. This initialization script will be sent to configure the DialServ device before making an outgoing connection, or receiving an incoming connection.

connection_wait_time=*time*

The amount of time, in seconds, to wait for a connection made from the DialServ. Incoming TCP connections will be severed if a connection cannot be made in this time.

See also

- revert: The **revert dialserv** command reverts the settings configured by this command.
- set profile: The **set profile profile=dialserv** associates the Dialserv port profile with the port.
- show: The **show dialserv** command shows the current Dialserv settings in a Digi device.

set dirp

Purpose

Configures Device-Initiated RealPort (DIRP) and displays current DIRP settings. A normal RealPort session/connection consists of a driver running on Windows, Unix, or Linux to connect to the Digi device to initiate RealPort sessions. This type of RealPort connection requires the driver to have the IP address of the Digi device, and that the Digi device has a static IP address. However, there are situation when such isn't possible, typically because the Digi device is set up to use DHCP, in which case, the IP address can change. Or, the Digi device may be behind a firewall, and cannot be accessible from the Windows, Unix, or Linux driver's side.

Device-Initiated RealPort changes which side of the connection creates the first connection for RealPort. Instead of the driver initiating the connection, the Digi device initiates the connection to the Windows, Unix, or Linux driver. Such a connection resolves both the DHCP and "behind the firewall" issues.

Note The driver running on the PC must support Device-Initiated RealPort. The only driver that does so is the Windows RealPort driver. Digi Linux and Unix drivers do not support Device-Initiated RealPort.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-network=read** to display settings, and **set permissions s-network=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Device-Initiated RealPort connections

```
set dirp [range=1-20]
  [state={on|off}
  [host={hostname|ip address}]
  [port=tcp port number]
  [retry=connect retry value in seconds]
  [encrypt={on|off}]
```

Display current Device-Initiated RealPort connection settings

```
set dirp
```

Options

range=1-20

Allows you to specify up to **20** independent Device-Initiated RealPort connections for use by the Digi device.

state={on|off}

Enables or disables Device -Initiated RealPort.

host={*hostname*|*ip address*}

The hostname or IP address of the Windows system that is running a RealPort driver in which they want the device to connect to.

port=*tcp port number*

The tcp port number that should be connected to on the Windows system specified by the **host** option.

retry=*connect retry value in seconds*

How often the Digi device should attempt to create the connection to the Windows system.

encrypt={on|off}

Determines whether Encrypted RealPort should be used (**on**) or not used (**off**) when the connection is made to the Windows system that is running the RealPort driver.

Examples

```
#> set dirp range=1 state=on host=WindowsServer1.digi.com port=8771 retry=30
encrypt=off
```

See also

- revert: The **revert dirp** command reverts the settings configured by this command.
- set realport
- show: The **show dirp** command shows the current device-initiated RealPort settings in a Digi device.

set dnsproxy

Purpose

Enables the DNS Proxy feature on the Digi device. The DNS Proxy feature permits DNS client hosts to communicate with this Digi device as if it were a DNS Server. The DNS Proxy will forward the DNS client's request to one of the DNS servers configured in its network settings. The device relays the response from the actual DNS serverto the requesting client when it is received by the DNS Proxy. Note that the DNS Proxy does not cache the actual detailed client requests nor the responses received from the DNS servers. Rather, it simply acts as a request/response relay agent between the DNS clients and servers.

The DNS Proxy cycles through the DNS servers configured in the Digi device. DNS client requests are identified by the client's IP address and the unique Query ID in the DNS request message. For each new DNS client request (new Query ID), the DNS Proxy uses the first DNS server in its list of DNS servers. If the client retries the same request (same Query ID), the DNS Proxy recognizes that retry message and either sends the retry request to the same DNS server as the previous request for this client, or it moves to the next DNS server in its list of DNS servers. The DNS Proxy feature determines when to retry the same DNS server, or move to the next DNS server, according to the “retries” (request retries per DNS server) option. The DNS Proxy itself does not perform unsolicited retries of DNS client requests.

Note The DHCP Server feature on the Digi device can be configured to use the DNS Proxy feature. For more information, see the **set dhcpserver** command and the **dnsproxy={on|off}** option.

The content of the DNS server list can be dynamic. For example, when DNS server IP addresses are received from a mobile service provider's network, they are added to the DNS server list of this Digi device server. Those DNS server IP addresses may or may not be configured when the DHCP Server offers a lease to a DHCP client. As a result, the DHCP client might have no DNS servers provided to it in the lease; as a result, domain name resolution might fail for that client. A significant benefit of the DNS Proxy feature is that the DHCP Server can offer its own IP address as a DNS server in the client lease, and the DNS Proxy forwards DNS requests and responses as stated above. Since the DHCP protocol does not allow a DHCP Server to force an unsolicited DNS server list update to its clients, the DNS Proxy feature provides an indirect method that makes such updates effective for the client.

Note The DHCP server will not offer its own IP address to its DHCP clients in the lease if the DNS Proxy is disabled.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-dnsproxy=read** to display settings, and to **set permissions s-dnsproxy=rw** to display and configure them. See set permissions for details on setting user permissions for commands.

Syntax

```
set dnsproxy [state={enabled|disabled}]
  [maxentries=16-1024]
  [idlettl=10-120]
  [retries=0-4]
```

```
{default: 1}
[onmaxentries={replacelru|discardnew}]
```

Options

state={enabled|disabled}

Enables or disables the DNS Proxy service.

maxentries=16-1024

Request cache size maximum. Specifies the maximum number of DNS client request records the DNS Proxy maintains concurrently in its cache. A large cache consumes more system resources than a small cache. However, if the maximum cache size is too small, new DNS client requests may be quietly discarded until the cache has room to add new client request records, or existing cache entries may be replaced by the new requests. If a large number of concurrent DNS client lookups is anticipated, configuring a larger maximum cache size is recommended. See also the **onmaxentries** option. The default is **256** entries.

idlettl=10 - 120

The period of time, in seconds, that a DNS client request remains in the DNS Proxy cache before it is deleted. This is a period of idle time, during which neither a DNS client request retry is received by the DNS Proxy, nor a DNS server response is received by the DNS Proxy, for a specific DNS client request. A shorter **idlettl** value results in resources being used more efficiently by the DNS Proxy, since the client request cache is reduced in size and the request buffers are released more quickly for future use for other DNS client requests. The default is **20** seconds.

retries=0-4

Request retries per DNS server; the number of retries using the same DNS server, for a specific DNS client request that is being retried (retransmitted) by the DNS client. There is always one “try” but the number of retries is configurable. The default is **1**.

onmaxentries={replacelru|discardnew}

Handling new client requests when the maximum number of client request entries is already being serviced; that is, the request cache is full.

replacelru

Remove the least recently used (LRU) client request entry from the cache, and add an entry for the new client request.

discardnew

Discard (ignore) new client requests until existing client requests have expired, allowing new requests. Silently discard the new client request, and do this for all future new requests until one or more entries have expired and been removed from the request cache.

The default is **replacelru**.

Examples

```
#> set dnsproxy state=enabled retries=2

#> set dnsproxy
```

```
DNS Proxy Configuration :

   state          : enabled
   maxentries     : 256
   idlettl        : 20
   retries        : 2
   onmaxentries   : replacelru
```

See also

- revert: The **revert dnsproxy** command reverts the settings configured by this command.
- set dhcpserver
- show: The **show dnsproxy** command shows the current DNS proxy settings in a Digi device.

set ekahau

Purpose

Configures Ekahau Client™ device-location software in a Digi Connect wireless device. The Ekahau Client feature provides integrated support for Ekahau's Wi-Fi device-location solution on wireless-enabled Digi devices. Ekahau offers a complete access point vendor-independent real-time location system for wireless LAN devices that is capable of pinpointing wireless LAN devices such as the Digi devices, laptops, PDAs, or other intelligent Wi-Fi enabled devices. The solution provides floor-, room- and door-level accuracy of up to 3.5 feet (1 m). The patented Ekahau positioning technology is based on simple signal-strength calibration maps, and enables customers to fully leverage an existing wireless LAN infrastructure without any need for proprietary hardware components.

Please visit www.ekahau.com for additional information, including free evaluation licenses for the Ekahau Positioning Engine and Ekahau Site Survey software products.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ekahau=read** to display Ekahau client settings, and **set permissions s-ekahau=rw** to display and configure Ekahau client settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Ekahau settings

```
set ekahau [state={on|off}]
  [id=device id]
  [poll_rate=seconds]
  [protocol={tcp|udp}]
  [port=port]
  [server={hostname|ip address}]
  [password=string]
  [name=string]
```

Display current Ekahau settings

```
set ekahau
```

Options

state={on|off}

Enables or disables the Ekahau Client feature.

The **id**, **name**, and **server** options must be set before you can set **state** to **on**.

id=*device id*

A numeric identifier for the Digi device, used internally by the Ekahau Positioning Engine for device tracking over time. This identifier should be unique for each Digi device being located on the network. It must be configured before the device will allow the **state** option to be set to **on**.

poll_rate=*seconds*

The time in seconds between each scan or wireless access points and communication with the server.

Once the Ekahau Client is enabled (**state=on**), every time the Digi device scans the network, it is disassociated with the access point (AP) providing its network connectivity. In addition, during the time, or scanning interval, set by the **poll_rate** option, it will not be receiving or transmitting wireless packets. This could lead to packet loss. Set the **poll_rate** as slow as acceptable in the application where the Digi device is being used.

The default is **5** seconds.

protocol={tcp|udp}

Specifies whether to use TCP or UDP as the network transport. The default is **tcp**.

port=*port*

The network port to communicate on. In the default Ekahau configuration, port **8548** is used for TCP, and port **8549** for UDP. This setting must be configured before the device will allow **state** to be set to **on**.

server={*hostname*|*ip address*}

The hostname or IP address of the Ekahau Positioning Engine. The maximum length of this option is **50** characters. The default is **8548**.

password=*password*

A password to authenticate with the server. The maximum length of this option is **50** characters. The default password for Digi and the Ekahau Positioning Engine is **Llama**.

name=*device name*

A descriptive name to identify the Digi device to users. The maximum length of this option is 50 characters. This name must be configured before the device will allow **state** to be set to **on**.

Examples

Set identifiers

```
#> set ekahau id=1 server=myepe.domain.com name=”Tracked Device 1"
```

Enable Ekahau Client

```
#> set ekahau state=on
```

See also

- revert: The **revert ekahau** command reverts the settings configured by this command.
- show: The **show ekahau** command shows the current Ekahau Client settings in a Digi device.
- For additional information, including free evaluation licenses for the Ekahau Positioning Engine and Ekahau Site Survey software products, see the Ekahau website at www.ekahau.com.

set ethernet

Purpose

Configures, adjusts, and displays Ethernet communications options.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-ethernet=read** to display Ethernet communications options, and **set permissions s-ethernet=rw** to display and configure Ethernet communications options. See set permissions for details on setting user permissions for commands.

Syntax

Configure Ethernet communications options

```
set ethernet [interface=interface name]
  [speed={auto|10|100}]
  [duplex={auto|half|full}]
  [mdimode={auto|mdi|mdix}]
```

Display Ethernet communications options

```
set ethernet
```

Options

interface=*interface name*

The name of the Ethernet network interface being configured, such as **eth0**, **eth0_1**, and so on.

speed={auto|10|100}

Configures the Ethernet speed the Digi device will use on the Ethernet network. Specify an appropriate setting for your Ethernet network, which can be one of the following:

auto

The device senses the Ethernet speed of the network and adjusts automatically.

10

The device operates at 10 megabits per second (Mbps) only.

100

The device operates at 100 Mbps only.

The default is **auto**.

If one side of the Ethernet connection is using **auto** (negotiating), the other side can set the Ethernet speed to whatever value is desired. Or, if the other side is set for **100** Mbps, this side must use **100** Mbps.

duplex={auto|half|full}

The mode the Digi device uses to communicate on the Ethernet network. Specify one of the following:

half

The device communicates in half-duplex mode.

full

The device communicates in full-duplex mode.

auto

The device senses the mode used on the network and adjusts automatically.

The default is **half**.

If one side of the Ethernet connection is using auto, the other side can set the duplex value to whatever is desired. If one side uses a fixed value (for example, half-duplex), the other side has to use the same.

mdimode={auto|mdi|mdix}

The connection mode for the Ethernet cable.

auto

Enables Auto-MDIX mode, where the required cable connection type (straight through or crossover) is automatically detected. The connection is configured appropriately without the need for crossover cables to interconnect switches or connecting PCs peer-to-peer. When it is enabled, either type of cable can be used and the interface automatically corrects any incorrect cabling. For this automatic detection to operate correctly, the **speed** and **duplex** options must both be set to **auto**.

mdi

The connection is wired as a Media Dependent Interface (MDI), the standard wiring for end stations.

mdix

The connection is wired as a Media Dependent Interface with Crossover (MDIX), the standard wiring for hubs and switches.

Examples

Configure 100 Mbps Ethernet speed

```
#> set ethernet speed=100
```

See also

- set network to configure network communications options.
- revert: The **revert network** command reverts the settings configured by this command.
- show: The **show ethernet** command shows the current Ethernet settings in a Digi device.

set failover

Purpose

The IP network failover feature provides a dynamic method for selecting and configuring the default gateway for the Digi device server. IP network failover uses of a set of rules and link tests to determine whether a particular network interface can be used to communicate with a specified destination. The user configures these rules, link tests and the priority order of the interfaces.

IP network failover can support the use of Ethernet, Wi-Fi and Mobile (cellular) network interfaces. The available interfaces vary among different Digi products.

IP network failover maintains a network interface list, ordered by the configured failover interface priority, and containing information on the state of the network interface and recent success or failure of the link tests for that interface. The failover status for a network interface is one of the following:

- **1 - Responding**: The interface is **Up** and configured in the system. It is currently responding to the link tests. This interface is suitable for use as the default gateway.
- **2 - Up**: The interface is **Up** and configured in the system. Its status has not been determined by the link tests, or no link tests are configured. This interface may be suitable for use as the default gateway.
- **3 - Not Responding**: The interface is **Up** and configured in the system. However, it is not currently responding to the link tests, and the number of consecutive test failures has reached the threshold number configured in the IP network failover settings. This interface may be suitable for use as the default gateway.
- **4 - Down**: The interface is **Down** or not configured in the system. However, it is not currently responding to the link tests. This interface is not suitable for use as the default gateway.
- **5 - Unknown**: The interface is **Unknown** (does not exist) in the system. This interface is not suitable for use as the default gateway.

The number shown above for each status value, indicates the priority of that status, used by failover in selecting the interface to use as the default gateway. Status priority 1 is the most suitable for use, with lower priorities considered suitable if there are no interfaces at the highest priority.

When any network interface changes status, the interface list is examined for the interface that has the highest status priority, nearest the start of the list. The highest priority interface with a **Responding** status is used as the default gateway. If no interface is marked **Responding** then the highest **Up** interface is used, and so on.

When IP network failover performs a link test, it adds a temporary static host route to the destination IP address for the link test, using the network interface that the link test is configured to test. The static host route is removed when the link test completes. whether successfully or in failure. Users should be careful to avoid manually configuring static host routes to any of the failover link test destinations, as such host routes may interfere with IP failover's link testing. Static IP routes are configured by the **set forwarding** command that configures the IP forwarding settings. See set forwarding.

The **set network** command's **gwpriority** (gateway priority) option provides a simpler method for selecting the default gateway. However, if IP network failover is properly configured and enabled, it

overrides any **gwpriority** value that is set. For a description of this non-failover gateway priority selection and information on how to configure it, see set network.

For IP Network Failover status and statistics, use the **display failover** command, or see the web interface under **Administration > System Information > IP Network Failover**.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-net-failover=read** to display IP network failover settings, and **set permissions s-net-failover=rw** to display and configure IP network failover settings. See set permissions for details on setting user permissions for commands.

Syntax

General options

```
set failover [state={off|on}]
  [fallback={off|on}]
  [prioritylist=list]
```

Configure IP network failover settings for a network interface

Note **if** can be used as an abbreviation for **interface**.

```
set failover [interface=interface name]
  [ifstate={off|on}]
  [testtype={none|ping|tcp}]
  [maxfailures=1-255]
  [interval=10-3600]
  [retryinterval=10-3600]
  [norespinterval=10-3600]
```

ICMP ping link test options

```
set failover [pingdest1=IPv4 address]
  [pingdest2=IPv4 address]
  [pingcount=1-10]
  [pinginterval=time]
```

TCP connection link test options

```
set failover [tcpdest1=IPv4 address]
  [tcpport1=1-65535]
  [tcpdest2=IPv4 address]
  [tcpport2=1-65535]
  [tcpconntimeout=time]
```

Display current IP network failover settings

```
set failover
```

Options

General options

state={off|on}

The IP network failover state; enables or disables the IP Network Failover feature in the Digi device.

fallback={off|on}

Enables or disables fallback to the non-failover default gateway priority method: The fallback option is used if a default gateway cannot be configured by IP network failover. Failure to configure a default gateway could occur if one or more interfaces are not enabled (on) for IP Network Failover use, or if the enabled interfaces are not up or do not have a gateway associated with them.

prioritylist=*list*

A comma-separated list of network interface names in priority order, used by failover to determine the default gateway. The default gateway is used to route IP packets to an outside network, unless controlled by another route. Default is **mobile0,eth0**. An empty list value means “Use the default.”

A network interface may have a static gateway configured for it, or it may obtain a gateway from DHCP or other means when the interface is configured. The first interface in this list that supplies a gateway will be used as the default gateway. The default gateway may change as interfaces connect and disconnect, and as failover link tests determine that an interface is providing the desired IP packet routing to a remote network destination.

Options for IP network failover for a particular network interface

ifstate={off|on}

The state for the IP network failover for a particular interface in the priority list of interface names, such as **eth0**.

testtype={none|ping|tcp}

The type of link test for this interface.

none

No link tests will be used for the network interface. Since no link tests are run, failover is only aware of the **Up** or **Down** status of the interface. This limited status information may affect the use of this interface as the default gateway.

ping

ICMP ping link test. For a description of this test, see the ICMP ping link test options.

tcp

TCP connection link test. For a description of this test, see the TCP connection link test options.

maxfailures=1-255

The number of consecutive link test failures before a **Not Responding** status is reported and a failover action may be taken. The default is **3**.

interval=10-3600

The time interval in seconds between the end of a successful link test and the start of the next link test for the network interface. This interval is used only after a successful test. The default is **240**.

Shorter intervals verify the link more often, but they also increase the packet traffic over the network interface being tested. The frequency of tests should be considered carefully for network connections

such as mobile (cellular) connections, which may be expensive, depending on the service plan in effect with your mobile service provider.

retryinterval=10-3600

The time interval in seconds between the end of a failed link test and the start of the next link test for the network interface. This interval is used after a failed test, but only until the "not responding consecutive failures" threshold (the **maxfailures** option) has been reached. The default is **240**.

A possible strategy is to configure a shorter retry interval than the success interval, to more quickly test the network connection to determine whether it is truly not working or there was just a transient test failure. Determining the validity of the link helps IP network failover determine whether it is necessary to reconfigure the default gateway.

norespinterval=10-3600

The time interval (N) in seconds between the end of a failed link test and the start of the next link test for the network interface. This interval is used after a failed test, but only after the "not responding consecutive failures" threshold (**maxfailures** option) has been reached. The default is **240**.

ICMP ping link test options

The ping test sends ICMP Echo Request packets to the configured destination IP address. If an ICMP Echo Reply is received (ping reply), the link test has successfully demonstrated that the network interface can be used to communicate with the specified destination.

pingdest1=*IPv4 address*

The primary, or first, destination to ping. The destination must be a valid IPv4 address. If this option is not specified, no primary destination link test will be attempted.

pingdest2=*IPv4 address*

The secondary, or second, destination to ping. The destination must be a valid IPv4 address. If this option is not specified, no Secondary Destination link test will be attempted.

pingcount=1-10

The maximum number of ping requests to send for a ping link test. When a reply is received, the ping test ends successfully and does not continue to send ping requests. If no ping reply is received after Send Count ping requests have been sent, the link test ends in failure. The default is **5**.

pinginterval=1-10

The time interval in seconds between sending ping requests during a ping link test. The ping tests sends a ping request. If no ping reply is received before the Send Interval expires, another ping request is sent. The default is **5**.

TCP connection link test options

The TCP Connection Test attempts to establish a TCP connection to the configured destination IP address and port number. If a connection is successfully established, or if the remote host actively rejects (resets) the connection attempt, the link test has successfully demonstrated that the network interface can be used to communicate with the specified destination. If a TCP connection is successfully established, it is immediately closed.

tcpdest1=*IPv4 address*

The primary, or first, destination to which to establish a TCP connection. The Primary TCP Port is used as the port to which the test connects at the Primary Destination. The destination must be a valid IPv4 address. If the destination is left empty, the device does not attempt a Primary Destination link test.

tcpport1=1-65535

The destination TCP port to use to connect to the Primary Destination address. Default is 80.

tcpdest2=*IPv4 address*

The secondary, or second, destination to which to establish a TCP connection. The Secondary TCP Port is used as the port to which the test connects at the Secondary Destination. The destination must be a valid IPv4 address. If the destination is left empty, the device does not attempt a Secondary Destination link test.

tcpport2=1-65535

The destination TCP port to use to connect to the Secondary Destination address. The default is **80**.

tcpconntimeout=10-60

The timeout in seconds to wait for a TCP test connection to be established or actively refused. The TCP test fails if the connection attempt times out. The default is **30**.

Examples

Determine whether IP network failover needs to occur. This command sets a ping test of interface **eth0** at an interval of 10 seconds.

```
#> set failover interface=eth0 ifstate=on testtype=ping interval=10
```

Next, set the **retryinterval** option, which sets how long to wait before sending the request again, in this case, ping, and the **norespinterval** option, which sets (how long to wait for the ping response, for the destination **192.168.250.1**, and enable IP network failover.

```
#> set failover interface=eth0 retryinterval=10 norespinterval=10
pingdest1=192.168.250.1 state=on
```

These commands set up a ping test, sending a request every 10 seconds to the destination **192.168.250.1**. When ping test fails, the gateway for the interface will change, according to the table in the **display failover** output for this device:

```
  Priority  Interface       Status              Gateway         State  Tests
     1      mobile0         2 (up)              10.0.0.1         on      0
     2      eth0            2 (up)              10.30.1.1        on      0
```

See also

- display failover
- info icmp
- info tcp
- revert The **revert failover** command reverts the settings configured by this command.

- set forwarding
- The **set network** command handles non-failover gateway priority management. See set network.
- show: The **show failover** command shows the current IP network failover settings in a Digi device.

set forwarding

Purpose

Configures IP routing, or forwarding of IP datagrams, between network interfaces. IP routing must be enabled to allow the Network Address Table (NAT) and port forwarding features to work properly.

The **set forwarding** command enables addition of static route entries to the IP routing table. Static routes instruct the device to route packets for a known destination host or network, to (through) a router or gateway different from the default gateway. They explicitly define the next “hop” from a router for a particular destination. This is sometimes necessary to communicate with hosts on a different subnet than the Digi device, or to provide a more direct or efficient route to such hosts than may be provided by using the default gateway or other routes.

Required privileges

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-router=read** to display IP forwarding settings, and **set permissions s-router=rw** to display and set forwarding settings. See set permissions for details on setting user permissions for commands.s

Syntax

Enable or disable IP forwarding:

```
set forwarding ipforwardingenabled={on|off}
```

Add a static route entry

Note that this command cannot be used to add, change or delete a *default* gateway entry; the default gateway is managed internally by the device.

To add a static route entry, all of the options must be entered.

```
set forwarding
  staticrouteindex={1-16}
  action=add
  enabled={on|off}
  net=destination ip address
 mask=subnet mask
 gateway=ip address
 metric={1-16}
  interface=interface name
```

Change existing static route entries

The **staticrouteindex** and **action** parameters are required for all static route entry management commands.

```
set forwarding
  staticrouteindex={1-16}
  action=change
 [one or more additional static route options, listed above]
```

```
 [mask=subnet mask]
 [gateway=ip address]
 [metric={1-16}]
  [interface=interface name]
```

Delete a static route

```
set forwarding
  staticrouteindex={1-16}
  action=delete
```

Enable a static route table entry

To enable a static route entry, all of the static-route options must be specified (on this or a previous command) and valid. That is, an entry can be enabled only if it is completely valid.

When a new entry is added or a change is made to a static route entry and that entry is enabled, the change is applied immediately to the IP routing table. If a static route entry is disabled or deleted, it is removed immediately from the IP routing table.

Display IP forwarding settings

```
show forwarding
```

Display the current active IP routing table

```
display route
```

Options

ipforwarding={on|off}

Enables or disables IP forwarding.

on

Enables IP forwarding.

off

Disables IP forwarding.

staticrouteindex={1-16}

Specifies which of the 8 static route table entries is to be acted upon.

action={add|change|delete}

Specifies the action to be performed on the selected static route table entry.

add

Add a new entry.

change

Change one or more options of an existing entry.

delete

Delete an entry, resetting all its options to defaults (empty).

enabled={on|off}

Enables or disables a static route table entry.

on

Enables an entry. All its options must be specified and valid to enable an entry. The enabled entry is immediately added to the device's IP routing table.

off

Disables an entry. If the entry was previously enabled and added to the device's IP routing table, that entry is immediately removed from the IP routing table.

net=*destination ip address*

Specifies the IP address of destination network or host to which the static route applies. This static route table entry defines how packets will be routed by the device when they are sent to the destination network or host.

mask=*subnet mask*

The subnetwork mask to be used for the static route, which is used in conjunction with the destination IP address. A subnetwork mask of **255.255.255.255** indicates that the destination IP address is a specific host rather than a network.

gateway=*ip address*

The IP address of the gateway for this static route. When the device routes packets that are destined for the specified destination IP address (host or network), those packets are sent to this gateway as their first “hop.”

metric={1-16}

Specifies the metric, or the “cost” to reach the destination. This is the “distance” in terms of number of “hops” for the routed packets to get to the destination.

interface=*interface name*

Specifies the name of the local network interface through which routed packets are sent for this static route. Valid interface names are: **eth0**, **ppp2**, **ppp4**, **vpn0**, **vpn1**, **vpn2**, **vpn3**, **vpn4**. To view the current list of interface names, enter a **display netdevice** command. For devices that support PPP interfaces, the actual PPP interface name may vary.

Examples

Enable IP forwarding

```
#> set forwarding ipforwarding=on
```

Add a static route entry

This command adds a static route for packets destined to hosts on the **10.10.0.0** network. These packets are sent through the **eth0** interface to a local router **10.30.1.1**. In this example, the device on which this static route is added has an IP address of **10.30.1.188**, and the router **10.30.1.1** is on its local subnetwork.

```
#> set forwarding staticrouteindex=1 action=add enabled=on net=10.10.0.0
mask=255.255.0.0 gateway=10.30.1.1 metric=2 interface=eth0
```

See also

- display netdevice
- display route displays current routing information.
- set nat
- revert: The **revert forwarding** command reverts the settings configured by this command.
- show: The **show forwarding** command shows the current IP forwarding settings in a Digi device.
- In the online help for the web interface, the topic **Configuration > Network Configuration > IP Forwarding Settings**.

set geofence

Purpose

Configures a Digi device for use with a Global Positioning System (GPS) and geofencing application. When a GPS device is present and configured for use in the Digi device's system, up to 16 geofences can be defined. A supported GPS receiver must be configured for use by the device. A geofence is defined by entry and exit radii around a latitude-longitude coordinate pair. GPS geofence options define perimeters around a point such that moving into, out of, or being outside of the perimeter trigger an alarm. Static position options define the latitude and longitude coordinates for the Digi device. Alarms can be configured based on the GPS device's relative position compared to the geofence: for exiting a fenced area, entering a fenced area, or being located outside of the fenced area. Alarm options set whether alarms will be reported to the Digi device's event log, an SNMP server, or via SMTP-based e-mail.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-gps-geofence=read** to display settings, and **set permissions s-gps-geofence=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure geofence settings

```
set geofence [index={1..16}]
  [state={enabled|disabled}]
  [name=name]
  [hdop_max=2.000000-30.000000]
  [latitude=-90.000000-90.000000]
  [longitude=-180.000000-180.000000]
  [entry_radius=10-10000000]
  [exit_radius=10-10000000]
  [outside_update_interval=30..86400, 0 = disabled]
  [email_recipient=email recipient]
  [email_ccrecipient=email ccrecipient]
  [email_subject=email subject]
  [email_body=email body]
  [email_from=email from]
  [priority={normal|high}]
  [primary_smtp_server=primary SMTP server]
  [secondary_smtp_server=secondary SMTP server]
  [log_entry_event={enabled|disabled}]
  [log_exit_event={enabled|disabled}]
  [email_entry_event={enabled|disabled}]
  [email_exit_event={enabled|disabled}]
  [trap_entry_event={enabled|disabled}]
  [trap_exit_event={enabled|disabled}]
  [log_update_outside={enabled|disabled}]
  [email_update_outside={enabled|disabled}]
```

```
[trap_update_outside={enabled|disabled}]
[gps_data_in_email_body={enabled|disabled}]
```

Display current geofence configuration settings

```
set geofence
```

Options

index={1..16}

Numeric identifier for the geofence. You can define up to **16** geofences.

state={enabled|disabled}

Indicates whether this geofence configuration should be activated. Setting this option to “enabled” tests the GPS unit’s position against the geofence specified on the **index** option.

name=*name*

A name for the geofence; this name appears in geofence alerts, whether they are sent by e-mail, logged in the event log, or issued as SNMP traps.

hdop_max=2.000000-30.000000

The maximum tolerated horizontal dilution of precision (HDOP), in degrees, that is allowed for reporting a geofence event. When the reported HDOP is greater than this value, fence event log reports, SNMP traps, and e-mail reports will not be sent. HDOP tolerances vary by GPS receiver. A lower number indicates more positional precision.

latitude=-90.000000-90.000000

Defines the latitude component of the geofence center, in degrees.

longitude=-180.000000-180.000000

Defines the longitude component of the geofence center, in degrees.

entry_radius=10-10000000

The entry radius, in meters, is the distance from the center of the fence for entry. That is, if the device is less than this distance from the defined center, an entry event has occurred.

exit_radius=10-10000000

The exit radius, in meters, is the distance from the center of the fence for exit. That is, if the device is more than this distance from the defined center, an exit event has occurred. This is also the distance used to determine if the device is outside of the fence for update events.

outside_update_interval=30..86400, 0 = disabled

The location update interval, in seconds, specifies the amount of time to wait between reporting that the device is outside of the geofence. This interval applies to event log, SNMP, and e-mail reports.

email_recipient=*email recipient*

The email address of the recipient of the geofence report e-mail.

email_ccrecipient=*email ccrecipient*

The email address of the carbon copy (CC:) recipient of the geofence report e-mail.

email_subject=*email subject*

The subject line that will appear on the geofence report e-mail.

email_body=*email body*

Body text for the e-mail. For example, **Device X has left the fenced area**.

email_from=*email from*

The email (return) address of the originator of the geofence report e-mail.

priority={normal|high}

The priority of the e-mail. Normal and high priority can be specified. High priority e-mails will display differently on some systems.

primary_smtp_server=*primary SMTP server*

The IPv4 address of the primary SMTP (e-mail) server.

secondary_smtp_server=*secondary SMTP server*

The IPv4 address of the secondary SMTP server. This server will be used if an e-mail cannot be sent to the primary SMTP server.

log_entry_event={enabled|disabled}

Specifies whether a log entry will be written when device has entered the geofence defined by the geofence center, and entry radius.

log_exit_event={enabled|disabled}

Specifies whether a log entry will be written when the device has left the geofence defined by the geofence center, and exit radius.

email_entry_event={enabled|disabled}

Specifies whether an e-mail will be sent to the defined recipients via the configured SMTP servers when the device has entered the geofence defined by the geofence center and entry radius.

email_exit_event={enabled|disabled}

Specifies whether an e-mail will be sent to the defined recipients via the configured SMTP servers when the device has left the geofence defined by the geofence center and exit radius.

trap_entry_event={enabled|disabled}

Specifies whether an SNMP trap will be sent to the defined SNMP servers when device has entered the geofence defined by the geofence center, and entry radius.

trap_exit_event={enabled|disabled}

Specifies whether an SNMP trap will be sent to the defined SNMP servers when the device has left the geofence defined by the geofence center, and exit radius.

log_update_outside={enabled|disabled}

A log entry will be written when the device is outside of the geofence defined by the geofence center, and exit radius. Log entries will be written at the interval defined by the **outside_update_interval** option.

email_update_outside={enabled|disabled}

An e-mail will be sent to the defined recipients via the configured SMTP servers when the device is outside of the geofence defined by the geofence center, and exit radius. E-mails will be sent at the interval defined by the **outside_update_interval** option.

trap_update_outside={enabled|disabled}

Specifies whether an SNMP trap will be sent to the defined SNMP servers when the device is outside of the geofence defined by the geofence center, and exit radius. SNMP traps will be sent at the interval defined by the **outside_update_interval** option.

gps_data_in_email_body={enabled|disabled}

Specifies whether the current location of the Digi device should be included in the geofence e-mail.Send out the current position in the e-mail, in addition to, or instead of **email_body**.

Examples

This example writes an entry into the device's event log when the device leaves the fenced area, 75M from the center of 44.1, -93.5.

```
#> set geofence index=1 state=enabled name=test_fence hdop_max=30.0
latitude=44.1 longitude=-93.5 entry_radius=50 exit_radius=75
log_exit_event=enabled
```

See also

- display gps to display the current position of the GPS device and information about the reading.
- revert: The **revert geofence** command reverts the settings configured by this command.
- set profile
- show: The **show geofence** command shows the current geofence settings in a Digi device.

set gpio

Purpose

Configure General Purpose I/O (GPIO) pins and displays current GPIO pin settings. In normal operation, the GPIO pins are used for the serial **CTS**, **DCD**, **DSR**, **DTR**, and **RTS** pins. **set gpio** allows you to use these GPIO pins for different purposes.

Default serial signal settings for GPIO pins

The default serial signal settings for the GPIO pins on a Digi device are as follows. Depending on the device, there are five or nine GPIO pins.

Pin Number	Default Serial Signal	Signal Direction
GPIO 1	DCD	Input
GPIO 2	CTS	Input
GPIO 3	DSR	Input
GPIO 4	RTS	Output
GPIO 5	DTR	Output
GPIO 6	TXD	Output
GPIO 7	RXD	Input
GPIO 8	TXD for port 2	Output
GPIO 9	RXD for port 2	Input

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-gpio=read** to display GPIO pin settings, and **set permissions s-ethernet=rw** to display and configure GPIO pins. See set permissions for details on setting user permissions for commands.

Syntax

Configure GPIO pins

```
set gpio range={1-n} mode={serial|input|output}
```

Display current GPIO pin settings

```
set gpio [range={1-n}]
```

Options

range={1-n}

Used to specify the index of the GPIO pin to manipulate, where ***n*** is the maximum number of GPIO pins on the device.

mode={serial|input|output}

The mode of operation of the GPIO serial pin.

serial

Indicates normal serial operation.

input

Allows input of GPIO signals. This is used in conjunction with alarms to trigger emails or SNMP traps indicating a particular signal change.

output

Allows output of GPIO signals. Currently, output of GPIO signals is not supported in the command-line interface. The web interface can be used to toggle the output of GPIO signals between high and low.

The default is **serial** for all GPIO pins.

Examples

Changing the operation of the GPIO signal pins

The following command changes GPIO pins 1-5 to allow input of GPIO signals.

```
#> set gpio range=1-5 mode=input
```

See also

- display gpio
- revert. The **revert gpio** command reverts the settings configured by this command.
- send for details on setting up alarms that issue email messages or SNMP traps when GPIO pins change.
- set alarm to set alarms based on GPIO pin state changes.
- show: The **show gpio** command shows the current GPIO pin settings in a Digi device.

set group

Purpose

Creates and manage user groups. **set group** performs the following actions:

- Adds a group. A maximum of 32 groups can be defined.
- Removes groups.
- Changes group configuration attributes.
- Displays group configuration attributes.

To apply a common set of user settings to more than one user, you may want to create a group with the required settings, and then associate that group with multiple users. If a user is a member of one or more groups, the user's effective permissions are the maximum of the permissions of the user and all of the groups to which the user belongs.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the group settings for the line on which they are logged in: **set permissions s-group=r-self**.
- For a user to display the group settings for any line: **set permissions s-group=read**.
- For a user to display and set the group settings for the line on which they are logged in: **set permissions s-group=rw-self**.
- For a user to display the group settings for any line, and “set group” settings for the line on which the user is logged in: **set permissions s-group=w-self-r**.
- For a user to display and set the group settings on any line: **set permissions s-group=rw**.

See set permissions for details on setting user permissions for commands.

Default permissions

When you create a new group, it has no permissions.

Syntax

Add a group

```
set group add id=number newname=string
```

Remove a group

```
set group remove {id=range|name=string}
```

Change group configuration attributes

```
set group {id=range|name=string} [newname=string]
[commandline={on|off}] [menu={none|index|name}] [defaultaccess=
{none|commandline|menu}]
```

Display group configuration attributes

```
set group {id=range|name=string}
```

Display group configuration attributes for all groups

```
set group
```

Options

add

Add a group. New groups are created with no permissions. Up to **32** groups can be defined.

remove

Remove groups.

id=*range*

Specifies the ID or range of IDs of the groups to be acted on.

name=*string*

Specifies the name of the group to be acted on.

newname=*string*

Specifies a new group name.

commandline={on|off}

Specifies whether the users in the group are allowed to access the command line of the device.

on

Users can access the command line interface.

off

Users can not access the command line interface.

The default is **on**.

menu={none|index|name}

Specifies whether the users in the group are allowed to access the custom menu interface of the device and defines the custom menu that the users will have displayed.

none

Users are not allowed to access the custom menu interface.

index

Users are allowed to access the custom menu interface and display the custom menu at the specified index.

name

Users are allowed to access the custom menu interface and display the custom menu using the specified name.

The default is “none.”

defaultaccess={none|commandline|menu}

Specifies the default access method and interface that users in the group will be given upon logging into the device. Note that the specified interface must be enabled for the group and have a valid menu if specified.

none

The group has no default access to the device and the users are not allowed to access either the command line interface or the custom menu interface without explicitly specifying the access method.

commandline

The users will be displayed and given access to the command line interface assuming the group has command line access rights enabled.

The default is **commandline**.

Examples

Add a new group

```
#> set group add newname=gurus id=4
```

Remove group 7

```
#> set group remove id=7
```

Set a new group name

```
#> set group id=4 newname=gurus
```

Set a group with command line access rights

```
#> set group id=4 commandline=on defaultaccess=commandline
```

See also

- User Models and User Permissions in Digi devices
- newpass
- revert: The **revert auth** command reverts the settings configured by **set group**.
- set menu
- set permissions

- set user
- show: The **show group** command shows the current user group settings in a Digi device.

set host

Purpose

Configures a name for the device, also known as a host name, or displays the current host name for the device.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-host=read** to display the current host name, and **set permissions s-host=rw** to display and set the host name. See set permissions for details on setting user permissions for commands.

Syntax

Configure a host name for the device

```
set host name=name
```

Display the current host name

```
set host
```

Options

name=name

The name for the device. If provided, this host name is placed in the **DHCP Option 12** field when the device is configured as a DHCP client (by **set network**) and requests IP configuration from a DHCP server. This is an optional setting that is only used when DHCP is enabled.

The name can be up to **32** characters long and can contain any alphanumeric characters, and can also include the **_** (underscore) and **-** (hyphen) characters.

Examples

```
#> set host name=hq_gateway
```

See also

- revert: The **revert host** command reverts the settings configured by this command.
- show: The **show host** command shows the current host name in a Digi device.

set hostlist

Purpose

Adds or removes entries from the host list. For DialServ, the host list provides a means to map a phone number (in the local name field) to a network destination, (in the **resolves_to** field).

When accessing a device by name, the Digi device attempts to locate the name in the host list. When a match is found, the host name is mapped to the alias. Typically, this host list is used as a first means of locating the destination address before using the domain name system (DNS).

Each host list entry consists of a local name string which resolves to a destination. The destination can be either an IP Address or Fully Qualified Domain Name (FQDN). Creating several entries in the host list allows a many-to-one mapping of multiple host names to a single destination, as well as a one-to-many mapping of a host name to multiple destinations. The one-to-many mapping allows a fail-over option; that is, a connection to the **resolves_to** name for the first host match in the list is attempted. If that connection attempt fails, the **resolves_to** name for the next match in the host list is used.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-hostlist=read** to display the current host list, and **set permissions s-hostlist=rw** to display and configure the host list. See set permissions for details on setting user permissions for commands.

Syntax

```
set hostlist [{add|remove}]
  local_name=host address or name as known locally
 resolves_to=host address or name that will be resolved
```

Options

{add|remove}

Adds or removes an entry in the host list.

local_name=*host address or name as known locally*

A local name string that maps to a destination.

resolves_to=*host address or name that will be resolved*

The destination to which the **local_name** value resolves. This destination can be either an IP Address or Fully Qualified Domain Name (FQDN).

See also

- revert: The **revert hostlist** command reverts the settings configured by this command.
- show: The **show hostlist** command shows the current host-list settings in a Digi device.

set ia

Purpose

Configures selected Digi devices to support special parsing and bridging of Industrial Automation (IA) protocols. For example, it enables the Digi device to function as a Modbus/TCP to serial Modbus Bridge. Command options allow for configuring:

- Protocol details of serial-port connected devices
- Protocol details of network-based masters
- Destination tables and route entries within the tables that define how protocol requests are forwarded to one of many slaves for a response.

Supported IA protocols include Modbus/RTU and Modbus/ASCII on the serial port or encapsulated within TCP/IP or UDP/IP, and Modbus/TCP transported by either TCP/IP or UDP/IP.

To displaying current IA settings, use the **show** command, as demonstrated in the Syntax section of this description.

For more information on Industrial Automation, see the Industrial Automation and Modbus knowledge base articles and technical notes on the Digi Support site: www.digi.com/support

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-ia=read** to display the current IA settings, and **set permissions s-ia=rw** to display and configure IA settings. See set permissions for details on setting user permissions for commands.

Syntax

There are several variants of syntax for the **set ia** command, depending on whether the command is being used to configure serial port-connected devices, network-based masters, or destination tables and route entries within destination tables. These syntax descriptions and their option descriptions are presented separately.

Configure serial-port connected devices—"set ia serial"

```
set ia serial=range [serial options] [modbus options]
```

```
[serial options]:
  type={master|slave}
  table=1..16
  protocol={modbusrtu|modbusascii}
  messagetimeout=100-99999 ms
  slavetimeout=10-99999 ms
  chartimeout=3-99999 ms
  idletimeout={0=disabled|1-99999 seconds}
  priority={high|medium|low}
```

```
[modbus options]:
  errorresponse={on|off}
  broadcast={on|off|replace}
```

```
  fixedaddress={auto|1-255}
  rbx={off|half}
```

To set the baud rates for the port, see set serial.

To enable IA protocols, set the serial port profile to **ia**. See set profile.

Configure network-based masters—"set ia master"

```
set ia master=range
 state={on|off}
  active={on|off}
  type={tcp|udp}
  ipport=ip port
 table=1..8
  protocol={modbusrtu|modbusascii|modbustcp}
  messagetimeout=100-99999 ms
  chartimeout=3-99999 ms
  idletimeout={0=disabled|1-99999 seconds}
  priority={high|medium|low}
```

Configure destination tables and route entries—"set ia table"

```
set ia table=range [table options]
  [route=range [route options]]
```

```
[table options]:
  state={on|off}
  name=string
 addroute=route index
 removeroute=route index
 moveroute=from_route_index,to_route_index
```

```
[route options]:
  active={on|off}
  connect={active|passive}
  protaddr=protocol address range
 type={discard|ip|mapto|nopath|serial|zigbee}
  protocol={modbusrtu|modbusascii|modbustcp}
  port=serial port
 transport={tcp|udp}
  address={ip address|dns name|XBee or ZigBee MAC address}
  ipport=ip port
 replaceip={on|add|sub|off}
  mapto=protocol address
 slavetimeout=10-99999 ms
  chartimeout=3-99999 ms
  idletimeout={0=disabled, 1-99999 seconds}
  reconnecttimeout=0-99999 ms
```

Display current IA settings

To display current IA settings, use the **show** command instead of a **set ia** command with no options:

```
show ia all
```

Options

Options for serial-port connected devices—"set ia serial"

serial=*range*

Specifies that the serial settings apply to the specified serial port or range of serial ports. The default is port **1**, and on a single-port device, entering serial is the same as **serial=1**.

[**serial *options***]

The serial settings, which include:

state={on|off}

Enables the IA serial settings. Setting to off risks the configuration being deleted.

active={on|off}

To temporarily stop processing the serial port, set **active** to **off**. The configuration will remain valid and saved.

type={master|slave}

Defines whether the serial device attached is acting as a master or a slave.

table=1..16—applies to master only

Defines which table is used to route messages to their destination. This option applies only to master-attached devices.

protocol={modbusrtu|modbusascii}

The protocol being used by the serial device. The serial protocol also affects the implied incoming network master on TCP and UDP ports **2101**.

modbusrtu

Modbus/RTU – 8-bit binary per Modbus specification at www.modbus-ida.org.

modbusasscii

Modbus/ASCII – 7-bit ASCII per Modbus specification at www.modbus-ida.org.

modbustcp

Modbus/TCP: or "Open Modbus" per Modbus specification at www.modbus-ida.org. Can be enabled with UDP/IP as well as TCP/IP.

messagetimeout=100-99999 ms—applies to master only

When messages are received from remote clients, this option defines the time to allow the message to be answered. This includes both the queuing and slave response delays, and this should be set to slightly less than the timeout of the remote client. After this time, the Digi device assumes the remote client no longer wants a response. The range is **100** to **99999** milliseconds. The default is **2500** milliseconds.

slavetimeout=10-99999 ms—applies to slave only

After all bytes of the message have been sent to the slave device, this is the time to wait for the first byte of a response. Note that the serial shift times are not included within this timeout. The range is **10** to **99999** milliseconds. The default is **1000** milliseconds.

chartimeout=3-99999 ms—applies to master or slave

After a first byte is received, this is the time to wait for additional bytes to either consider the message complete or to be aborted. The default is protocol-specific, but usually **50** milliseconds.

idletimeout={0=disabled|1-99000 seconds}

The device aborts a connection on the implied incoming master sockets after the remote client has been idle for this time. The time is saved in seconds, and the best use for this timeout is to speed up fault recovery. For example, many wide-area networks can suffer shutdowns without the Digi device detecting it. Using the idle timeout speeds up detection of lost TCP connections. The range is **1** to **99999** seconds. The default is **5** minutes.

priority={high|medium|low}

Normally messages are processed in a fair round-robin scheme. This becomes unfair when one master acts as many – for example opening 16 TCP sockets to talk to 16 slaves contrasted to a second master using a single TCP socket to talk to 16 slaves. In this situation, the device assumes it has 17 masters and in effect the first master will have 16 requests answered for every one the second master succeeds in getting answered. This option can be used to adjust the handling of serial master requests. For example, set the serial master to **high** and the network masters to **medium**. This option's effectiveness depends on the protocol behavior. Therefore, while some Modbus systems find it useful, it has no effect on most Rockwell protocols.

high

A high-priority master can get up to 50 percent of the bandwidth – of course you cannot have too many high-priority masters. All high-priority masters with queued messages get one message serviced before any low or medium priority masters get any service.

medium

If a high-priority master has queued messages, then one medium-priority master gets one message serviced before all the high-priority masters are offered service again. If only medium-priority masters exist (which is the default setting), all masters are serviced in a round-robin manner.

low

Low-priority masters only get service when no high- or medium-priority master has messages to service.

The default is **medium**.

[modbus options]

The configuration options specific to the Modbus protocol, which include:

errorresponse={on|off}

Controls behavior for common run-time errors such as no response from the slave device. By default, **errorresponse=off** for serial Modbus protocols, since most masters assume no response when errors occur. Having this option off also actively filters out returning Modbus exception codes **0x0A** and **0x0B** from remote Modbus/TCP slaves.

broadcast={on|off|replace}

Specifies how to handle incoming requests with a slave address set to the broadcast value. For Modbus, this is **0**. The default is to replace **0** with **1**. This default was established to overcome the fact that many Modbus/TCP clients always send requests to unit ID zero (**0**) and do not want this request treated as a broadcast.

on

Tells the Digi device to send requests as broadcast to the destination device(s) and not expect any response message.

off

Tells the Digi device to throw away the broadcast request.

replace

Changes a broadcast request to a normal request by replacing the **unit ID 0** with a value of **1**.

fixedaddress={auto|1-255}

Used to override the Modbus protocol address (**unit ID**) with a fixed address.

auto

When set to **auto**, the protocol address will not be overwritten.

1-255

Setting it to a fixed number from **1-255** forces this value to be used for all Modbus requests.

The default is **auto**.

rbx={off|half}

Enables the serial slave driver to handle Report-By-Exception (XMIT) writes between polls.

The Modbus/RTU serial slave driver has a Report-By-Exception (or XMIT) handler. If this handler is enabled, the Digi device pauses between slave polls to receive potential “master requests” initiated by the slave. Just as with a serial master-attached configuration, the Modbus/RTU **slave address** is used with the IA route table to determine the remote destination.

Behavior is assumed to be half-duplex, and the exact behavior of the slave device after a collision where both units try to send a request at the same time is unpredictable. The Digi device normally ignores the XMIT request and treats its own request as a timeout.

The **fixedaddress** option must be set to a specific value, not **auto** or **0**. All other values

are assumed to be XMIT transactions. This enables any Modbus message seen from the slave and *not* equaling the **fixedaddress** to be treated as unsolicited master request. The message will be routed by the table to other destinations. For example, if **fixedaddress=1**, it is assumed that any network request for slave **1** goes to the serial device, and any message from the serial device marked as from slave **1** is assumed a response to a network request. However, any messages from a slave marked as from slave **2-255** are assumed as master requests instead, and handled through the normal routine table.

Options for network-based masters—"set ia master"

master=*range*

Specifies the index of the network master to which the master options apply.

state={on|off}

Enables the IA network master settings.

active={on|off}

Enables or disables the network listener that accepts network connections.

type={tcp|udp}

Defines whether the incoming connection is TCP (connected) or UDP (unconnected). The default is **tcp**.

For cellular connections, using UDP/IP can cut 40% to 60% from your monthly bill.

ipport=*ip port*

Defines the UDP or TCP port on which to listen for protocol messages. Modbus/TCP defaults to TCP port **502**.

table=1..8

Defines which table is used to route messages to their destination. This option applies only to master-attached devices.

protocol={modbusrtu|modbusascii|modbustcp}

The protocol used for the connection.

modbusrtu

Modbus/RTU – 8-bit binary per Modbus specification at www.modbus-ida.org.

modbusascii

Modbus/ASCII – 7-bit ASCII per Modbus specification at www.modbus-ida.org.

modbustcp

Modbus/TCP: or "Open Modbus" per Modbus specification at www.modbus-ida.org. Can be enabled with UDP/IP as well as TCP/IP.

messagetimeout=100-99000 ms

When messages are received from remote clients, this is the time to allow the message to wait to be answered. This includes both the queuing and slave response delays, and this should be set to slightly less than the timeout of the remote client. After this time, the Digi device assumes the remote client no longer wants a response. The range is **100** to **99000** milliseconds. The default is **2500** milliseconds.

chartimeout=3-99000 ms

After a first byte is received, this is the time to wait for additional bytes to either consider the message complete or to be aborted. The default is protocol-specific, but usually **50** milliseconds.

idletimeout={0=disabled|1-99000 seconds}

The device aborts a connection after the remote client has been idle for this time. The time is saved in seconds, and the best use for this timeout is to speed up fault recovery. For example, many wide-area networks can suffer shutdowns without the Digi device detecting it. Using the idle timeout will speed up detection of lost TCP connections. The range is **1** to **99999** seconds. The default is **5** minutes.

priority={high|medium|low}

Normally messages are processed in a fair round-robin scheme. This becomes unfair when one master acts as many – for example opening 16 TCP sockets to talk to 16 slaves contrasted to a second master using a single TCP socket to talk to 16 slaves. In this situation, the device assumes it has 17 masters and in effect the first master will have 16 requests answered for every one the second master succeeds in getting answered. This option can be used to adjust the handling of serial master requests. For example, set the serial master to **high** and the network masters to **medium**. The effectiveness of this option depends on the protocol behavior. Therefore, while some Modbus systems will find it useful, it has no effect on most Rockwell protocols.

high

A high-priority master can get up to 50 percent of the bandwidth – of course you cannot have too many high-priority masters. All high-priority masters with queued messages get one message serviced before any low or medium priority masters get any service.

medium

If a high-priority master has queued messages, then one medium-priority master gets one message serviced before all the high-priority masters are offered service again. If only medium-priority masters exist (which is the default setting), then all masters are serviced in a round-robin manner.

low

Low-priority masters only get service when no high- or medium-priority master has messages to service.

The default is **medium**.

Options for destination tables and route entries—"set ia table"

The destination table and routes are used by the incoming master connections to select which one of many potential slaves a request is to be answered by.

table=*range*

Selects one of eight possible tables in which to look up forwarding information.

*[**table options**]*

The configuration options specific to the destination table, which include:

state={on|off}

Enables the IA destination table settings.

name=*string*

A useful name for the destination table. Default names are **table1**, **table2**, and so on. This option gives you the option to rename the table for convenience. Tables are still handled internally by number.

addroute=*route index*

Inserts a new route at this index of the table. If an existing route occupies this index, it is pushed up to a higher index.

removeroute=*route index*

Destroys the route at this index in the table.

moveroute=*from_route_index,to_route_index*

Moves the destination route from one route index to another.

route=*range*

Specifies the index of the route in this table to which the settings apply.

[**route options**]

The configuration options specific to the route table entries in the destination table, which include:

active={on|off}

Enables or disables the route in the table.

connect={active|passive}

Defines whether the Digi device attempts immediately to connect to a remote device (**active**), or waits and only connects on demand (**passive**). The default is **passive**.

protaddr=protocol address range

Defines the range of protocol addresses to be forwarded to this destination entry in the table. You can specify a single address or an inclusive range. The permitted values are defined by the protocol. The table is scanned from the first route index to the last, stopping at the first route with the appropriate protocol address. Duplicates or overlapping ranges can exist, but the route with the lowest index will be used.

type={discard|ip|mapto|nopath|serial|zigbee}

Defines the type of destination for this route.

discard

Messages destined for this route entry are discarded without error.

ip

Messages destined for this route entry are forwarded to the entered IP address. If you enter the IP address as **0.0.0.0**, the Digi device's IP address is used to fill in the IP address, and the **replaceip** function is applied. For example, if the IP address is **0.0.0.0**, the Digi device's IP address is **143.191.23.199**, and the protocol address of the message is **45**, then the remote IP address used will be **143.191.23.45**.

mapto

Messages destined for this route entry are reevaluated as if having the protocol address configured within this entry.

nopath

Messages destined for this route entry are returned to sender with a protocol-defined error message.

serial

Messages destined for this route entry are forwarded to a serial port.

zigbee

Messages destined for this route entry are encapsulated on a Digi XBee wireless network, with the targeted node defined by the IEEE MAC address of the XBee module encoded in the address setting. This option is supported on XBee-enabled products only.

protocol={modbusrtu|modbusascii|modbustcp}

The protocol used for the connection. See www.modbus-ida.org for information on specifications.

modbusrtu

Modbus/RTU – 8-bit binary per Modbus specification.

modbusascii

Modbus/ASCII – 7-bit ASCII per Modbus specification.

modbustcp

Modbus/TCP – or “Open Modbus” per Modbus specification.

Can be enabled with UDP/IP as well as TCP/IP.

port=*serial port*

Defines the serial port for the route table entry. This option applies only if the route **type=serial**.

transport={tcp|udp}

Applies only if the route **type=ip**. Defines whether the outgoing connection is TCP (connected) or UDP (unconnected). The default is **tcp**.

address={*ip address*|*dns name*|*XBee or ZigBee MAC address*}

Applies only if the route **type=ip**. The destination IP address of the entry.

ipport=*ip port*

Applies only if the route **type=ip**. The UDP or TCP port on which to listen for protocol messages. Modbus/ TCP defaults to TCP port **502**.

replaceip={on|add|sub|off}

Applies only if the route **type=ip**.Specifies whether and how the last octet of the IP address is replaced.

on

The protocol address is used to replace the last octet of the IP address. For example, if the table IP is **192.168.1.75** and the protocol address of this message is **23**, the message will be forwarded to the remote IP **192.168.1.23**.

add
sub

If the **add** or **sub** value is set, the protocol address is added or subtracted from the final octet of the IP address. In the above example, the result would be **192.168.1.98** or **192.168.1.52**, respectively.

off

The last octet of the IP address is not replaced.

The default is **off**.

mapto=*protocol address*

Used for destination entries of type **mapto**. This option defines the protocol address for which to reevaluate this message.

slavetimeout=10-99999 ms

After all bytes of the message have been sent to the slave device, this is the time to wait for the first byte of a response. The range is **10** to **99999** milliseconds. The default is **1000** milliseconds.

chartimeout=3-99999 ms

After a first byte is received, this is the time to wait for additional bytes to either consider the message complete or to be aborted. The default is protocol-specific, but usually **50** milliseconds. In TCP/IP network context, this can be thought of as the re-fragment time between packets.

idletimeout={0=disabled, 1-99999 seconds}

The connection is closed after no new messages have been forwarded to the remote slave (server) for this idle time. The time is saved in seconds. The range is **0** to **99999** seconds, where 0 means never close this connection. The default is **5** minutes.

reconnecttimeout=0-99999 ms

If the connection to the remote node fails and **connect=active**, this time is used to delay attempts to reconnect. The default is **2500** milliseconds.

Examples

Serial port configuration settings

The following **set ia** commands show the default serial-port configuration settings for the Digi Connect WAN IA:

```
set ia serial=1 active=on type=slave protocol=modbusrtu table=1
set ia serial=1 messagetimeout=2500 slavetimeout=1000 chartimeout=20
set ia serial=1 priority=medium idletimeout=0
set ia serial=1 errorresponse=off broadcast=replace fixedaddress=0
set ia table=1 route=1 active=on type=serial protaddr=0-32 port=1
```

Note that future Digi Connect WAN IA firmware releases will expose these configuration settings in the web interface.

Enable a Modbus/RTU serial master

To enable a Modbus/RTU serial master instead, change the configuration settings by entering the following **set ia** commands via Telnet. These **set ia** commands change the serial device type from slave to master. In addition, to handle cellular latency, a timeout value of **31** seconds or longer is needed. In addition, the first route, which forwards requests to the serial port, can be turned off.

```
set ia serial=1 type=master messagetimeout=31000
```

```
set ia table=1 route=1 active=off
```

Additional examples

For more examples of the **set ia c**ommand, see the application note *Remote Cellular TCP/IP Access to Modbus Ethernet and Serial Devices* available at www.digi.com/support.

See also

- revert: The **revert ia** command options revert any existing IA configuration settings.
- set profile. The **ia** port profile configures a serial port for controlling and monitoring various IA devices and PLCs.
- set serial
- show: The **show ia master**, **show ia serial**, and **show ia table** commands display current IA configuration settings in a Digi device.
- Industrial Automation and Modbus knowledge base articles and technical notes on the Digi Support site: www.digi.com/support.

set login

Purpose

Suppresses the user login for a Digi device.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-login=read** to display the login settings, and **set permissions s-login=rw** to display and configure login settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set login [suppress={on|off}]
```

Options

suppress={on|off}

Specifies whether the user login is suppressed for the Digi device.

on

Suppress login. The Digi device uses the one-user model, as described in User Models and User Permissions in Digi devices.

off

Do not suppress login. The Digi device uses the two-user model, as described in User Models and User Permissions in Digi devices.

See also

- revert: The **revert login** command reverts the settings configured by this command.
- show: The **show login** command shows the current user login settings in a Digi device.
- User Models and User Permissions in Digi devices

set mgmtconnection

Purpose

Configures or displays Device Cloud server connection settings. A Device Cloud server allows devices to be configured and managed from remote locations. These connection settings set up the connection to the Device Cloud server so, the Digi device knows how to connect to the server.

You can choose how your Digi device connects to and communicates with the Device Cloud server, through a device-initiated connection, a server-initiated connection, or a (device-initiated) timed connection, including paged connections. If Short Message Service (SMS) capabilities are enabled on how your Digi device, a paged connection is another means by which a device-initiated connection may be requested.

In a *device-initiated connection*, the Digi device attempts to reach the Device Cloud server to establish the connection. An advantage of the device-initiated connection is that it can be used on any network, whether the device has a public or private IP address, as long as the Device Cloud server is accessible on that network.

A *server-initiated connection* works the opposite way. The Device Cloud server opens a TCP connection and the Digi device listens for the connection. An advantage of a server-initiated connection is that the connection is only established when it is needed - this minimizes the overhead of maintaining a connection. A disadvantage is that Device Cloud server's device list will display the device as disconnected most of the time. In addition, server-initiated connections cannot be used if the device has a private IP address or is behind a NAT.

A *timed connection* is another form of a device-initiated connection. For a timed connection, the Digi device attempts to connect to the Device Cloud server at a configured, regular interval (period). If a connection to a Device Cloud server is already established, the timed connection will not be attempted. The next attempt for a timed connection will occur at the next scheduled interval.

A *paged connection* is another form of a device-initiated connection. This type of connection is initiated by an on-demand request, such as a Short Message (SM) received via a cellular modem (from a mobile service provider). The request message may specify the Device Cloud server with which the device should connect, or it may simply request that the device connect to the Device Cloud server that is configured in the Paged Device Cloud Connection settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-mgmtconnection=read** to display Device Cloud connection settings, and **set permissions s-mgmtconnection=rw** to display and set Device Cloud connection settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Device Cloud server connection settings

```
set mgmtconnection [svraddr[1-4]=string]
  [secidx[1-4]=index]
  [conntype={client|timed|serverinitiated|paged}]
  [connenabled={on|off}]
  [timedperiod=1-65535]
```

```
[timedoffset={immediate|oneperiod|randomtime}] [lkaupdateenabled={on|off}]
[clntreconntimeout={none|timeout]
[pagedoverrideenabled={on|off}]
```

Display Device Cloud server connection settings

```
set mgmtconnection
```

Options

svraddr[1-4]=*string*

Server address text. Used to specify one of eight possible Device Cloud server addresses. When the device server attempts to connect to the Device Cloud server, it tries the server addresses in this list in the order 1-4.

secidx[1-4]=*index*

Index into security settings. Used to link a server address with a Device Security entry.

conntype={client|timed|serverinitiated|paged}

Used to specify the connection type.

client

This is a client connection.

timed

This is a timed connection.

serverinitiated

This is a server-initiated connection.

paged

This is a paged connection. A paged connection is support for receiving a short message (SMS) that can request that the device connect to a Device Cloud server. The connection to Device Cloud is initiated by the client. A paged connection is a temporary, as-needed Device Cloud connection that is created based on a user action, for example, by calling a Python function, or sending a request-connect SMS command.

When enabled and a request is received to do so, the Digi device will initiate the connection to the Device Cloud server. A paged connection is initiated on demand when a request to connect is received from an external communication, such as a Short Message received via a mobile service provider. The external communication may specify the Device Cloud server with which the device should connect, or it may simply request that the device connect to the Device Cloud server that is configured in the Paged Device Cloud Connection settings.

Paged Device Cloud Connections provide emergency access to your Digi device, directing it to connect to the Device Cloud server so management or application operations may be performed.

A Paged Device Cloud Connection can be configured to disconnect an established connection to an Device Cloud server so the paged connection can be established instead, or it may configured to defer to a connection that is already established.

If paged Device Cloud connections are not enabled by this setting and the **connenabled** setting, paged connection requests will be refused if received via external communication. This setting and the **connenabled** setting controls whether or not paged Device Cloud connections will be permitted.

connenabled={on|off}

Whether this connection instance is enabled for use.

on

Enables this connection instance for use.

off

Disables this connection instance for use.

timedperiod=1-65535

For a timed connection, the time interval in minutes between the device server's attempts to connect to the Device Cloud server. If a device server is already in a connection to a Device Cloud server when the time interval expires, it will not start a new connection at that time. Rather, the device server will start a new timed period timer, and it will again check whether it needs to connect to the Device Cloud server when that new timer expires. The default is **5**.

timedoffset={immediate|oneperiod|randomtime}

For a timed connection, when the first timed connection to a Device Cloud server should be attempted after the Digi device boots.

immediate

Attempt to connect immediately.

oneperiod

Wait one full timed period, then attempt to connect.

randomtime

Wait some random interval of time, between 0 and the full timed period, then attempt to connect.

lkaupdateenabled={on|off}

In conjunction with a server-initiated connection, this option enables or disables a connection to a Device Cloud server to inform that server of the IP address of the device server. This permits the Connectware Manager to connect back to the device server, or to dynamically update a DNS with the IP address of the device.

on

Enables “last known address” connections to the Device Cloud server.

off

Disables “last known address” connections to the Device Cloud server.

clntreconntimeout={none|*timeout*}

Specifies the retry timeout interval, in seconds, for a last-known-address (LKA) update, if the LKA update fails. If an LKA update fails, the interval configured by this option is used as the amount of time to wait before attempting another LKA update. This option is used for both client-initiated and server-initiated connections. The keyword **none** turns off the retry timeout interval feature.

pagedoverrideenabled={on|off}

Controls how the Digi device handles user requests for a paged connection.

on

If a user requests a paged connection and an Device Cloud connection already exists, the current connection is dropped and a new connection is created.

off

The paged connection request is ignored.

The most common scenario is that paged and client-initiated connections use the same server URL, so if the Digi device is connected via any connection, it is connected, and this option should be off. Otherwise, the Digi device will unnecessarily drop its connection and reconnect if this option is on.

Examples

Set values for the client connection

```
#> set mgmtconnection connenabled=on conntype=client clntreconnecttimeout=50
```

Display current connection settings

```
#> set mgmtconnection
```

See also

- display dcloud
- revert: The **revert mgmtconnection** command reverts the settings configured by this command.
- set devicesecurity
- set mgmtglobal
- set mgmtnetwork
- show: The **show mgmtconnection** command shows the current Device Cloud server connection settings in a Digi device.
- The *Device Cloud User Guide*.
- The *Device Cloud Programming Guide*.

set mgmtglobal

Purpose

A Device Cloud server allows devices to be configured and managed from remote locations. This command is used to set or display the Device Cloud global settings, or revert the device ID to factory settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-mgmtglobal=read** to display settings, and **set permissions s-mgmtglobal=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Device Cloud global settings

```
set mgmtglobal [deviceid={hex string}]
  [revertdeviceid]
  [rcicompressionenabled={on|off}]
  [tcpnodelayenabled={on|off}]
  [tcpkeepalivesenabled={on|off}]
  [connidletimeout={none|timeout value}]
  [dataserviceenabled={on|off}]           (default=on}
  [dataserviceurl=valid url (path only)]  (default=/ws/device)
  [dataserviceport=0-65535]               (default=800
  [dataservicesecureport=0 - 65535]       (default=443)
```

Display Device Cloud global settings

```
set mgmtglobal
```

Revert the Device ID to factory settings

```
set mgmtglobal revertdeviceid
```

Options

deviceid={*hex string*}

Used to specify the device ID. The device ID is 32 hexadecimal digits, preceded by the characters **0x**.

rcicompressionenabled={on|off}

Configures whether RCI command and response text is compressed, when both are passed between the Digi device and the Device Cloud server. This compression primarily affects the size of the data passed when settings or state information are formatted as RCI and conveyed between device and server. Using compression on this RCI text can reduce the size of passed data, and, for cellular products, reduce the cost of reading and writing device settings.

When RCI compression is enabled, LIBZ compression is used on RCI command and response text when it is sent between device and server. The protocol used to manage and pass data between devices and Device Cloud, known as EDP, internally negotiates whether compression is applied. RCI compression is enabled, or **on** by default to reduce byte count and cost of sending data. As an example of savings, typical cellular router settings will compress to about 8% of its original size, which means that data can be sent in far fewer packets and less time, than when the uncompressed version of the same data is sent.

The default is **on**. The ability to turn off RCI compression is provided for technical support/troubleshooting purposes; for example, if you want to eliminate the possibility that this compression is causing a problem.

tcpnodelayenabled={on|off}

Configures whether use of the **TCP NODELAY** option is disabled by default for the Device Cloud connection between device and server, when configuring the device's TCP socket endpoint for that connection.

The ability to turn on the **TCP NODELAY** option is provided for technical support/troubleshooting purposes.The default is **off**. This default reduces the number of packets sent when the Device Cloud connection is established between device and server. While there is a very slight penalty in terms of added latency, that penalty is very small compared to the relative high latencies for cellular network communications. Reducing the packet count reduces the number of bytes exchanged over the cellular connection, which saves money. The typical start-up data count is reduced from about 7KB to 4KB just by disabling **TCP NODELAY**.

tcpkeepalivesenabled={on|off}

Enables or disables sending of TCP keep-alive packets over the client-initiated connection to the Device Cloud server, and whether the device waits before dropping the connection. The default is **on**.

TCP keep-alives are performed at the TCP protocol level. The application (in this case, the Device Cloud server) that is using that connection does not know anything about when the TCP keep-alives are sent or received. The TCP keep-alives simply serve to keep each end of the TCP connection aware that the connection is still viable, and intermediate network equipment (NATs in particular) is also made aware that the connection is still good.

connidletimeout={none|*timeout value*}

Enables or disables the idle timeout for the Device Cloud connection between device and server. Specifying **none** disables the idle timeout. Specifying a timeout value enables the idle timeout, which means the connection will be dropped, or ended, after the amount of time specified. The default is **on**. The minimum value is **300** and the maximum **43200**.

In contrast to TCP keep-alives, the timeout managed by the **connidletimeout** option is at the Device Cloud application level. The **connidletimeout** option provides a way for the connection to the Device Cloud server to be closed if no Device Cloud protocol data is sent or received for some period of time. This capability is particularly useful for server-initiated connections. When a user at the server side requests that a connection be established to a device, that user needs to explicitly terminate the connection when they are done with the device. This timeout permits a way to configure the device such that a “forgetful user” does not inadvertently leave the connection in place, which could cost money on a cellular connection if Connectware or TCP keepalives are enabled and transferred needlessly between device and server.

revertdeviceid

Reverts the device ID to factory settings. For example, if the device’s MAC address is **GG:HH:JJ:KK:LL:MM**, the device ID is set to **0x0000000000000000GGHHJJ****ffff****KKLLMM**.

Examples

Set the device id

```
#> set mgmtglobal deviceid=0x0123456789abcdef0123456789abcdef
```

See also

- display dcloud
- revert: The **revert mgmtglobal** command reverts the settings configured by this command.
- set devicesecurity
- set mgmtconnection
- set mgmtnetwork
- show: The **show mgmtglobal** command shows the current Device Cloud global settings in a Digi device.
- The *Device Cloud User Guide*
- The *Device Cloud Programming Guide*

set mgmtnetwork

Purpose

A Device Cloud server allows devices to be configured and managed from remote locations. **set mgmtnetwork** configures the network settings for the Digi device's connection to the Device Cloud server so the device knows how to connect to the server.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-mgmtnetwork =read** to display settings, and **set permissions s-mgmtnetwork =rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure Device Cloud network settings

```
set mgmtnetwork
  [networktype={modemppp|ethernet|802.11}] [connectionmethod=
{auto|none|mt|mtssl|mdh|proxy}] [proxyaddress=    address text]
  [proxyport=port]
  [proxylogin=login text]
  [proxypassword=password text]
  [proxypersistentconnection={on|off}]
  [mtrxkeepalive=time in seconds]
  [mttxkeepalive=time in seconds]
  [mtwaitcount=wait count]
  [mdhrxkeepalive=time in seconds]
  [mdhtxkeepalive=time in seconds]
  [mdhwaitcount=wait count]
  [sslvalidatepeer={on|off}]
```

Note **networktype=modemppp** is used for mobile networks (WiMAX and cellular).

Note For Device Cloud service, **connectmethod** must be set to **mt** or **mtssl**, since the other connection methods are not supported by Device Cloud.

Display Device Cloud network settings

```
set mgmtnetwork
```

Options

About keep-alive and wait-count settings on this command

The **set mgmtnetwork rxkeepalive** settings specify how frequently the device sends a keep-alive packet to the server if the Device Cloud connection is idle. The server expects to receive either data

messages or keep-alive packets from the device at this interval.

The **txkeepalive** settings specify how frequently the server sends a keep-alive packet to the device if the Device Cloud connection is idle. The device expects to receive either data messages or keep-alive packets from the server at this interval.

After the number of consecutive expected keep-alives specified on the w**aitcount** option are missed according to the configured intervals, the connection is considered lost and is closed by the device and server.

networktype={modemppp|ethernet|802.11}

The type of network to which this command applies.

modemppp

A modem PPP network.

Note **networktype=modemppp** and any settings associated with it apply to both cellular and WiMAX networks used by the Digi device.

ethernet

An Ethernet network.

802.11

An 802.11 network.

connectionmethod={auto|none|mt|mdh|proxy}

The firewall traversal method used by the protocol used to manage and pass data between devices and Device Cloud, known as EDP.

auto

Automatically detect the connection method.

none

No firewall; connect using TCP.

mt

Connect using TCP.

mtssl

Connect using an SSL connection to the Device Cloud server. This method offers for improved security of the connection and is the default connection method in current Digi device firmware.

mdh

Connect using HTTP.

proxy

Connect using HTTP over proxy.

proxyaddress=*address text*

The proxy host address when the connection method is **proxy**.

proxyport=*port*

The proxy host port when the connection method is **proxy**.

proxylogin=*login text*

The login string when the connection method is **proxy**.

proxypassword=*password text*

The proxy password when the connection method is **proxy**.

proxypersistentconnection={on|off}

Whether the device server should attempt to use HTTP persistent connections when the connection method is **proxy**. Not all HTTP proxies correctly handle HTTP persistent connections. The use of persistent connections can improve performance of the exchange of messages between the device server and the Device Cloud server, when that connection is HTTP/proxy. The reason for this is that the same HTTP connection can be reused for multiple consecutive HTTP requests and replies, eliminating the overhead of establishing a new TCP connection for each individual HTTP request/reply, then closing that connection when the request is complete.

on

The device server should attempt to use HTTP persistent connections.

off

The device server should not attempt to use HTTP persistent connections.

mtrxkeepalive=*time*

The transmit keep alive time when connection method is **mt** or **mtssl**, where ***time*** is the number of seconds to wait between sending keep-alive messages.

mttxkeepalive=*time*

When the connection method is **mt** or **mtssl**, the receive keep alive time, where ***time*** is the number of seconds to wait for a keep-alive message from the Device Cloud server before assuming the connection is lost.

mtwaitcount=*count*

When the connection method is **mt** or **mtssl**, used to specify the wait count where ***count*** is how many timeouts occur before the Digi device assumes the connection to the Device Cloud server is lost and drops the connection.

mdhrxkeepalive=*time*

When the connection method is **mdh**, used to specify the transmit keep alive timlocal_address={IP where ***time*** is the number of seconds to wait between sending keep-alive messages.

Important: It is recommended that this interval value be set as long as your application can tolerate to reduce the amount of data traffic.

mdhtxkeepalive=*time*

When the connection method is **mdh**, used to specify the receive keep alive time, where ***time*** is the number of seconds to wait for a keep-alive message from the Device Cloud server before assuming the connection is lost.

mdhwaitcount=*count*

When the connection method is **mdh,** used to specify the wait count, where ***count*** is how many timeouts occur before the Digi device assumes the connection to the Device Cloud server is lost and drops the connection.

sslvalidatepeer={on|off}

If **connectmethod=mtssl**, selects whether the server's certificate is required to be validated before the Digi device’s connection to the Device Cloud server is made.

on

The Device Cloud connection will only be allowed if the server certificate is validated. If it cannot be validated, the connection is not made. The recommended method of validating the server's certificate is by y installing a CA certificate for the signer or the server's certificate. Having the server's certificate installed as a “trusted peer” certificate is not recommended.

off

The Device Cloud connection is allowed without validating the server certificate.

Examples

Set instance 1 for proxy connection

```
#> set mgmtnetwork connectiontype=modemppp connectionmethod=proxy
proxyaddress=”What goes here?” proxyport=40002 proxylogin=”johnsmith”
proxypassword=”testpass” proxypersistentconnection=off
```

Set instance 2 for mdh connection

```
#> set mgmtnetwork connectiontype=ethernet connectionmethod=mdh
mdhrxkeepalive=100 mdhtxkeepalive=110 mdkwaitcount=15
```

Display current Device Cloud network settings

```
#> set mgmtnetwork
```

See also

- display dcloud
- revert: The **revert mgmtnetwork** command reverts the settings configured by this command.
- set devicesecurity
- set mgmtconnection
- set mgmtglobal
- show: The **show mgmtnetwork** command shows the current Device Cloud network settings in a Digi device.
- The *Device Cloud User Guide.*
- The *Device Cloud Programming Guide.*

set mobile

Purpose

Configures mobile (cellular) settings. The purpose of **set mobile** command and its settings differ for Digi cellular-enabled device with a CDMA or GSM cellular module.

Digi cellular-enabled devices with CDMA

For Digi cellular-enabled devices with CDMA modules, **set mobile** allows you to lock down on a specific technology, either 1xRTT, or EvDO, or allow the device to automatically select the technology.

For applications that require EvDO speeds to work properly, the **set mobile** command can lock down on EvDO, and will not make use of 1xRTT, even if it is available and EvDO is not.

For areas with spotty EvDO coverage, modules can be locked down to 1xRTT, to avoid the module trying to lock in on an EvDO signal.

Digi cellular-enabled devices with GSM

The Digi device may be equipped with one or two Subscriber Identity Module (SIM) cards. A SIM card contains the account information associated with a particular mobile service provider. All of the SIM card configuration options are stored individually for each SIM card.

The device uses the primary SIM first to try to establish a connection. If the connection is unsuccessful, it uses the secondary SIM instead. If it is also unsuccessful, the device tries the primary and then secondary SIMs again repeatedly. Several command options allow selection of which SIM is primary, and control over when to switch between SIMs.

For Digi cellular-enabled devices with GSM modules, **set mobile** allows you to select a specific band or technology. By specifying 3G bands, the module is effectively locked into high speed 3G bands, which is useful for applications that require higher speeds. For areas that have spotty 3G coverage, but good 2G coverage, 2G bands may be set to ensure that unit has optimal signal in this environment.

The GSM set mobile command also allows you to lock in a carrier by entering the mobile country code (MCC), and mobile network code (MNC). The module makes a best effort attempt to lock into the carrier. This helps prevent roaming charges.

Status and statistics for mobile communications

To display information and statistics for mobile/cellular communications, issue a **display mobile** command. To display SureLink statistics, issue a **display pppstats** command (see display pppstats

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display settings, and **set permissions s-ppp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure a Digi cellular-enabled device with a CDMA module

```
set mobile [provider=provider_name]
  [technology={automatic|1xrtt|evdo}]
```

Configure a Digi cellular-enabled device with a GSM module and one or two SIMs

```
set mobile [provider=provider_name]
  [carrier={automatic|mobile country code+mobile network code]
  [band={automatic|band[,band...]}
  where:
  band={850|900|1800|1900|3g_850|3g_900|3g_1900|3g_2100}|2g_only|3g_only]
  [sim_pin=string]
```

Note Gobi (aka Huawei EM680) can be either CDMA or GSM, depending on which provider is set.

Configure a Digi cellular-enabled device with a GSM module and two SIMs

```
set mobile index=1-2
  [set_primary]
  [register_time={1-65535 seconds|0=disable}]
  [connect_attempts={1-255|0=disable}]
  [switch_roaming={no|yes}]
  [switch_dropped={no|yes}]
  [switch_idle_time={1-65535 seconds|0=disable}]
  [switch_max_time={1-65535 seconds, 0=disable}]
```

Configure the integrated GPS receiver for any Digi cellular-enabled device

```
set mobile [gps_state={disabled|enabled_always|enabled_connected}]
  [gps_method={standalone|mobile_based|mobile_assisted}]
  [gps_count={1-999)|continuous}]
  [gps_time={1-255 seconds}]
  [gps_interval=1-65535 seconds]**
  [gps_accuracy=1-65535 meters]
```

** For Verizon service, this option value depends on the **gps_method** value. See the **gps_method** description.

Options

CDMA module-specific mobile settings

For CDMA modules (on the ConnectPort WAN only—Sierra Wireless 572X modules), options are:

provider=*provider_name*

The mobile service provider for the cellular modem. To display a list of supported providers, enter the **help set mobile** command.

technology={automatic|1xrtt|evdo}

The type of cellular technology used by the CDMA module.

automatic

The CDMA module can select the type of technology to use.

1xRTT

The CDMA uses 1xRTT technology (a 2G technology; slower),

evdo

The CDMA uses EvDO (a 3G technology; faster).

Configure this setting depending on applications that run in the Digi device and the type of mobile coverage in the area. For example, if an application streams video, and will only work at EvDO speeds, locking down on EvDO is desired. On the other hand, if EvDO coverage is spotty, the technology should be set to 1xRTT.

GSM module-specific mobile settings

provider=*provider_name*

The mobile service provider for the cellular modem. To display a list of supported providers, enter **help set mobile**.

index=1-2

The slot number of the SIM card. The *Quick Start Guide* for your product shows the default designations for primary and secondary SIM cards.

set_primary

Sets the SIM card specified on the **index** option to be the primary or preferred SIM to use to establish mobile connections.

SIM configuration options

Options to stop using one SIM card and switch to the next

These next options determine when a connection attempt is unsuccessful, at which point the Digi device should switch to the next SIM card to establish mobile connections.

register_time={1-65535 seconds|0=disable}

Causes the Digi device to switch from one SIM card to the other if the SIM has not registered with the mobile service provider after the specified number of seconds.

connect_attempts={1-255|0=disable}

Causes the Digi device to switch from one SIM card to the other if a connection could not be established after the specified number of connection attempts.

switch_roaming={no|yes}

Causes the Digi device to switch from one SIM card to the other if the SIM card is registered, but is roaming to another mobile service provider. Note that your mobile service provider may apply additional connection charges when roaming.

Options to disconnect a SIM card and return to primary SIM card

Once a connection has been successfully established with this SIM, these options determine when to end the connection and return to using the primary SIM.

switch_dropped={no|yes}]

Causes the Digi device to disconnect a SIM card when the connection is dropped or ended for any reason.

switch_idle_time={1-65535 seconds|0=disable}

Causes the Digi device to disconnect a SIM card if the connection is idle or no data has been received over the mobile link for the specified number of seconds.

switch_max_time={1-65535 seconds, 0=disable}

Connection timeout. Causes the Digi device to disconnect a SIM card if the connection has been established for the specified number of seconds. This option can be used to switch back to the primary SIM if the secondary SIM costs too much to use.

Options for carrier and band selection

carrier={automatic|*mobile country code+mobile network code*}

Specifies how the GSM module selects the mobile carrier, either automatically, or a particular carrier.

Mobile carrier selection allows the mobile device to be configured to use a specific mobile service only. The recommended and normal operation is for the mobile device to automatically find service with an available carrier. However, a manual selection can be configured to force the use of a particular carrier. Please be aware that use of a manual carrier selection can result in a significantly longer time interval for the unit to find service on the specified network. Both the mobile network and the mobile device (modem) may influence this behavior. Therefore it is recommended that the automatic selection be used wherever possible.

To scan for available carriers, enter a **display carriers** command.

WARNING! The scan for available carriers requires that the mobile PPP connection be terminated to perform the scan. A successful scan cannot be performed and completed if it is initiated over the mobile connection, since the scan procedure requires user interaction that is not possible after the mobile PPP connection has been terminated.

Note The mobile PPP connection is not automatically restarted when a carrier selection is configured. The configuration change will take effect when you restart the mobile connection or reboot the Digi device.

automatic

Configures the GSM module to select the carrier automatically.

mobile country code+mobile network code

The numeric concatenation of a country code and network code. This setting forces the radio to use a particular carrier. Setting the carrier to avoid roaming, or when coverage by a provider is spotty.

For example, this command locks down the carrier to T-Mobile using country code **310** (USA), and network code **26** (T-Mobile):

```
#> set mobile carrier=31026
```

Additional information regarding manual carrier selection

A specific carrier selection may be requested by entering it manually or selecting it from the carrier scan results list. However, the subsequent success of using that carrier depends on a number of factors that are outside the control of the Digi device server. These factors include, but are not limited to: the configuration stored in the SIM, your individual service agreement with your provider, and roaming agreements among service providers in your area. These factors are subject to change by the service providers, and you must configure the carrier selection accordingly. Please consult your service provider with questions or issues regarding your service.

Although multiple providers may offer service in your area, there may be agreements between them that disallow a mobile device using one provider's SIM, from connecting to and using another service provider's network. If you manually configure your carrier selection to use another provider's network, the connection attempts may either time out (because they are ignored by the provider's network) or you may notice that the connection attempts are actively rejected or disconnected due to authentication failures. In either case, the mobile connection will not be established successfully. To correct this problem, reconfigure the carrier selection as either Automatic or try another manual carrier selection that is compatible with your SIM's service account.

Note that the available carrier scan will identify all the service providers that are found in your area. However, that list may include service providers that will not allow access by your device, due to the factors stated above.

In summary, although you may request the use of a specific carrier, the decision to allow or deny access to that carrier's network is enforced by the network. That decision cannot be overridden by the carrier selection configuration of the Digi device server.

sim_pin=*string*

The PIN code for the SIM card. This option is applicable only for certain mobile service providers and does not need to be set if your SIM card does not have a PIN.

If your Digi device has dual SIM cards, each SIM may have its own PIN. Set it with this command:

```
set mobile index=n sim_pin=xxxx
```

band={automatic|*band*[,*band*...]}

Specifies mobile service frequency bands to be configured in the GSM cellular modem. Use the default selection, **automatic**, unless there is a reason to configure specific bands only.

The band selection can be done automatically by the GSM module, or can be locked down to a single band, 3G, or 2G (slower).

Note The mobile PPP connection is not automatically restarted when a band selection is configured. The configuration change will take effect will take effect on the next connection.

automatic

Enables automatic service band selection by the modem.

band={850|900|1800|1900|3g_850|3g_900|3g_1900|3g_2100|2g_only|3g_only}

Selects the individual service bands to be configured. Improper selection or combinations may result in a failure to establish a mobile connection. The bands are in MHz. Select one or more of the values shown above.

Band selection allows selecting a subset of the supported bands. Use this option to restrict bands to those that get the best signal. For example, a carrier could have both 850 and 1900 coverage, but if 1900 is better, the GSM module could be restricted to 1900. Or, if a lower frequency has better penetration into the a wiring closet, the band can be restricted to that frequency.

3g_850, **3g_900**, **3g_1900**, and **3g_2100** are for UMTS/HSDPA (3G cellular) bands, and are supported on ConnectPort WAN models only.

On 3G-capable modules, band selection allows for specifying whether 2G or 3G only is desired. The choices **2g_only** and **3g_only** select only 2G service or 3G service. Since not all cell modems support selection of specific bands, but they do support broader selection of **2G service only** or **3G service only**, these choices are available. For example, if you get better 2G coverage, and wish to restrict the module to 2G to get the best signal, select **2g_only**. Or, if application requirements dictate the use of 3G (for example, streaming video), select **3g_only**. Sometimes a 2G signal will be very good, and a bad 3G signal will be present. In this case, the cellular modem may attempt to use the 3G signal, because it is faster, even though it is poor. Being able to specify that the 2G signal should be used exclusively alleviates that issue.

Options for integrated GPS receiver

The following options configure the Global Positioning System (GPS) receiver that is integrated into the mobile module of the Digi device. These settings do not affect dedicated or external GPS devices.

gps_state={disabled|enabled_always|enabled_connected}

Enables or disables the GPS receiver.

disabled

The GPS receiver is not active.

enabled_always

The GPS receiver is always active.

enabled_connected

The GPS receiver is active while a mobile data connection is established. Use this setting if the GPS receiver is interfering with making data connections.

gps_method={standalone|mobile_based|mobile_assisted}

The method used to determine a position fix.

standalone

The GPS receiver determines a position without any assistance from the mobile network. It must obtain all necessary information from GPS satellites.

mobile_based (network assisted)

The GPS receiver obtains satellite almanac and ephemeris data from the mobile network. This may reduce the time to determine the first position fix.

mobile_assisted (network calculated)

The GPS receiver may send raw satellite data to the mobile network. The network calculates the position of the mobile device, using additional information available to the base station. This may increase the accuracy of position fixes.

Note Your provider may apply additional connection charges when network assistance or calculation is used.

gps_count={1-999)|continuous}

The number of position fixes. The GPS receiver can provide continuous position fixes, or stop after the specified number of fixes.

gps_time={1-255 seconds}

The maximum amount of time allowed to determine a position fix.

gps_interval=1-65535 seconds]

Interval between fixes; that is, the time between the start of one position fix to the start of the subsequent fix.

For Verizon service, the minimum value for the **gps_interval** option depends upon the selected value for the **gps_method** option:

gps_method:	gps_interval minimum:
standalone	1
mobile_based	30
mobile_assisted	1800

gps_accuracy=1-65535 meters

The preferred accuracy of a position fix. If this accuracy is not available in the time allowed, the best available position is provided.

Examples

```
#> set mobile provider=sprint_pcs technology=evdo
```

```
#> set mobile index=1 provider=cingular_orange register_time=60
```

```
#> set mobile gps_state=enabled_connected gps_method=standalone
```

See also

- display carriers
- display mobile
- revert: The **revert mobile** command reverts the settings configured by this command.
- set mobileppp
- set surelink
- show: The **show mobile** command shows the current mobile settings in a Digi device.

set mobileppp

Purpose

Configures Point-to-Point Protocol (PPP) settings used for mobile connections, or displays current mobile PPP settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display settings, and **set permissions s-ppp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set mobileppp index=1-2
  [state={enabled|disabled}]
  [auth_method={none|pap|chap|both}]
  [remote_address={IP address|negotiated}]
  [local_address={IP address|negotiated}]
  [address_mask=IP address mask]
  [default_gateway={yes|no}]
  [ipcp_dns_enabled={on|off}]
  [protocol_compression={on|off}]
  [address_compression={on|off}]
  [header_compression={on|off}]
  [lcp_keepalive={on|off}]
  [lcp_ka_quiet_time=10-86400 seconds]
  [lcp_ka_max_missed_replies=2-255; 0=ignore missed replies]
  [asyncmap=hex string]
  [chap_id=CHAP id]
  [chap_key=CHAP key]
  [pap_id=PAP id]
  [pap_password=PAP password]
  [mru=1500-2048]
  [mtu=1500-2048]
  [redial_attempts=0-100; 0=infinite]
  [redial_delay=1-64000]
  [rx_idle_timeout=0-86400; 0=off]
  [tx_idle_timeout=(0-86400; 0=off]
  [init_script="chat script"]
  [dial_script="chat script"]
  [login_script="chat script"]
  [n{1-4}=phone number]
  [proxy_arp={on|off}]
  [device_description="description text"]
  [passive={on|off}]
```

Options

index=1-2

The index number of the SIM used for the mobile PPP connection. This index number is required on dual-SIM devices.

state={enabled|disabled}

The state of the interface. The default is **disabled**.

auth_method={none|PAP|CHAP|both}

Determines whether authentication is required for mobile PPP connections and, if so, what kind.

none

The remote user does not require PPP authentication.

pap

Password Authentication Protocol (PAP) authentication is required.

chap

Challenge Handshake Authentication Protocol (CHAP) authentication is required.

both

Both CHAP and PAP authentication are required.

The default is **none**. CHAP authentication works between two Digi devices. CHAP will be negotiated to PAP for all other connections.

passive={on|off}

(***This option exists, but it's used for debugging (to select NDIS vs. PPP).***

Specifies whether the device server waits for the remote system to begin PPP negotiations, or can initiate PPP negotiations on its own.

on

The device server waits for the remote system to begin PPP negotiations.

off

The device server may initiate PPP negotiations.

The default is "**off**."

Do not set both sides of a PPP connection to "passive=on."

remote_address={*ip address*|negotiated}

The address of the peer at the other end of the mobile PPP connection. Either a specific address or the keyword "negotiated" can be specified; **negotiated** means that the address will be accepted from the peer. An IP address of all zeroes is equivalent to specifying the keyword **negotiated**.

local_address={*ip address*|negotiated}

The IP address of the local end of the mobile PPP connection. Enter either a specific address or the keyword **negotiated**; **negotiated** means that the address will be accepted from the peer. An IP address of all zeros is equivalent to specifying the keyword **negotiated**.

address_mask=*ip address mask*

The IP mask to apply to the address specified on the **remote_address** and **local_address** options. Specifying a specific IP address on the **remote_address** and **local_address** options modifies the meaning of the IP address for routing purposes. The default is **255.255.255.255**.

default_gateway={yes|no}

Selects whether to use the PPP interface as the default route. The default is **no**.

ipcp_dns_enabled={on|off}

Enables or disables the IPCP (PPP Internet Protocol Control Protocol) acquisition of DNS IP addresses. This option is enabled by default to preserve prior behavior.

protocol_compression={on|off}

Specifies whether the device server attempts to negotiate protocol compression on mobile PPP connections.

on

The device server attempts to negotiate protocol compression on mobile PPP connections.

off

The device server will not negotiate protocol compression.

The default is **on**.

address_compression={on|off}

Specifies whether the device server attempts to negotiate address compression on mobile PPP connections.

on

The device server attempts to negotiate address compression.

off

The device server does not attempt to negotiate address compression.

The default is **on**.

header_compression={on|off}

Specifies whether the device server attempts to negotiate IP protocol header compression on mobile PPP connections. This is commonly referred to as Van Jacobsen (VJ) header compression.

on

The device server attempts to negotiate IP protocol header compression.

off

The device server does **not** attempt to negotiate IP protocol header compression.

The default is **on**.

lcp_keepalive={on|off}

Specifies whether the Digi device sends Link Control Protocol (LCP) echo requests after a “quiet” interval, to test the PPP link and/or keep it alive. “Quiet” means not having received any bytes over the PPP link for a specified time interval, which is set by the **lcp_ka_quiet_time** option. In PPP networks that support LCP echoes, an LCP echo reply is returned by the remote end of the mobile PPP connection.

Even if LCP keepalives are disabled in this device (by **lcp_keepalive=off**), the device still replies to LCP echo request messages it may receive from the remote side of the mobile PPP connection by sending an LCP echo reply message. But the device itself will not originate any LCP echo request messages.

The options are:

on

The device server sends LCP echo requests after a configurable “quiet” interval, set by the **lcp_ka_quiet_time** option.

off

The device server does not send LCP echo requests.

lcp_ka_quiet_time=10-86400 seconds

Specifies the “quiet” interval, in seconds, after which the device server sends an LCP echo request. “Quiet” means not having received any bytes over the PPP link for the interval specified by this option.

lcp_ka_max_missed_replies={2-255|0=ignore missed replies}

Specifies how many consecutive echo replies may be missed before the device server disconnects the PPP link. A value of 0 (zero) specifies that the device server should not act on missed LCP echo replies by disconnecting the PPP link. Note that if bytes of any kind, LCP echo reply or otherwise, are received, the PPP link is deemed to be active, and the “missed LCP echo replies” count is reset to zero.

asyncmap=*hex string*

A mask for PPP connections that defines which of the 32 asynchronous control characters to transpose. These characters, in the range 0x00 to 0x1f, are used by some devices to implement software flow control. These devices may misinterpret PPP transmission of control characters and close the link. This mask tells PPP which characters to transpose.

The default is **FFFF**, which means transpose all 32 control characters. Any combination is valid. The following are the masks most likely used:

FFFFFFFF

Transpose all control characters.

00000000

Transpose none.

000A0000

Transpose **Ctrl-Q** and **Ctrl-S**.

chap_id=*CHAP id*

A character string that identifies the mobile PPP user using CHAP authentication. This is equivalent to a user or login name. The string must be 32 or fewer characters and must be recognized by the peer.

chap_key=*CHAP key*

A character string that authenticates the mobile PPP user using CHAP authentication. This is equivalent to a password. The string must be 16 or fewer characters and must be recognized by the peer.

pap_id=*PAP id*

A character string that identifies the mobile PPP user using PAP authentication. This is equivalent to a user (or login) name. The string must be 32 or fewer characters and must be recognized by the peer.

pap_password=*PAP password*

A character string that authenticates the mobile PPP user using PAP authentication. This is equivalent to a password. The string must be 16 or fewer characters and must be recognized by the peer.

mru=1500-2048

The maximum received unit (MRU), or frame size, in bytes, to be received from the other end of the PPP connection. This is a negotiated value. The default is **1500** bytes.

mtu=1500-2048

The maximum transmission unit (MTU), or frame size, in bytes, to use for this PPP outbound connection. For PPP connections, the MTU is negotiated, so enter 1500, the largest size device server permits the remote host to send. For PPP users, the range is **128** to **1500** bytes, and the default is **1500** bytes.

redial_attempts=0-100; 0=infinite

The number of times the firmware attempts to redial before giving up.

redial_delay=1-64000

The time, in seconds, to wait after an unsuccessful dial attempt.

rx_idle_timeout=0-86400; 0=off

The time, in seconds, after which if no data has been received over the link, the PPP connection is disconnected.

tx_idle_timeout=(0-86400; 0=off

The time, in seconds, after which if no data has been transmitted over the link, the PPP connection is disconnected.

init_script="*chat script*"

An initialization script, run once at interface startup. For example:

```
init_script="'' ATZ OK \c"
```

dial_script="*chat script*"

A dialing script, used any time a number is dialed for the interface. For example:

```
dial_script="'' ATDT\T CONNECT \c"
```

login_script="chat script"

A login script, used to log in to the remote system on the other end of the mobile PPP connection. For example:

```
login_script="ogin: <username> assword: <password>"
```

n*{1-4}=*phone number

Up to four phone numbers to dial to request a mobile PPP connection. The phone numbers are dialed sequentially.

proxy_arp={on|off}

When enabled, performs proxy ARP for the remote peer of the mobile PPP session, so that the peer can be made to appear on our local network. It performs that proxy ARP on the subnet to which the IP address assigned to that peer belongs.

device_description="*description text*"

An alternate string used for SNMP purposes. If this option value is set, it is used in the customized SNMP description for this interface. If not specified in the settings, the PPP device name, for example, **ppp4**, is used instead. This string allows for SNMP-produced information to fit better with network management software that prefers identification strings to be in a format, such as **mobile0**.

See also

- display pppstats displays status and activity information for a (PPP) link, including SureLink statistics.
- provision

- revert: The **revert mobileppp** command reverts the settings configured by this command.
- set mobile
- set surelink
- set ppp
- show: The **show mobileppp** command shows the current mobile PPP settings in a Digi device.

set nat

Purpose

Used to set or display Network Address Translation (NAT) and port/protocol forwarding settings.

Note that at this time, the only IP protocols for which protocol forwarding is supported are:

- Generic Routing Encapsulation (GRE, IP protocol 47)
- Encapsulating Security Payload (ESP, IP protocol 50, tunnel mode only).

Port forwarding is supported for the TCP and UDP protocols.

You can forward a single port or a range of ports. To forward a range of ports, specify the number of ports in the range using **pocount** option in the port forwarding entry.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to "**set permissions s-router=read**" to display settings, and **set permissions s-router=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Set NAT and port/protocol forwarding settings

```
set nat
  instance=1-8 (required for devices supporting more than one NAT instance)
  [ifname=public network interface name]
  [enabled={on|off}]
  [maxentries=64-1024]]
  [dmzenabled={on|off}
  [dmzip=ip address]
  [prenabled[1-4]={on|off}]
  [prnumber[1-4]={gre|esp}]
  [prtype[1-4]=type]
  [prip[1-4]=ipaddress]
  [poenabled[1-64]={on|off}]
  [poproto[1-64]={tcp|udp|ftp}]
  [pocount=[1-64]=number of ports in range, minimum 1]
   Note: must be 1 for "poproto=ftp"
  [poexternal[1-64]=number of ports in range, minimum 1]
  [pointernal[1-64]=number of ports in range, minimum 1]
  [poip[1-64]=ipaddress]
```

Display NAT and port/protocol forwarding settings

```
set nat
```

Options

instance=1-8

For Digi devices that support multiple NAT instances for different network interfaces, the NAT instance to which the **set nat** command applies. Required for devices supporting more than one NAT instance; that includes nearly all Digi products that support NAT.

ifname=p*ublic network interface name*

The name of the network interface for which NAT performs address and port translations. The list of interfaces available for NAT configuration varies according to the capabilities of your Digi device server model. For this device, valid interface names are:

mobile0, **vpn0**, **vpn1**, **vpn2**, **vpn3**, **vpn4**, **eth0**.

enabled={on|off}

Enables or disables NAT. Note that IP forwarding must be enabled by the **set forwarding** command for NAT to work.

on

Enable NAT.

off

Disable NAT.

dmzenabled={on|off}

Enables or disables DMZ forwarding to the IP address specified by the **dmzip=*ip address*** option.

DMZ forwarding allows specifying a single host, known as a DMZ server, on the private (internal) network that is available to anyone with access to the NAT public interface IP address, for any TCP- and UDP-based services that have not been configured. Services enabled directly on the Digi device server take precedence over, and are not overridden by, DMZ forwarding. Similarly, TCP and UDP port forwarding rules take precedence over DMZ forwarding. DMZ forwarding is effectively a lowest-priority default port forwarding rule that does not permit the same remapping of port numbers between the public and private networks, as is possible when using explicit port forwarding rules.

If enabled, the DMZ forwarding rule is used for incoming TCP and UDP packets from the public (external) network, for which there is no other rule. These other rules include explicit port forwarding rules or existing dynamic rules that were created for previous communications, be those outbound (private to public) or inbound (public to private). Also, the DMZ Forwarding rule is not used if there is a local port on the Digi device server to which the packet may be delivered. This includes TCP service listener ports as well as UDP ports that are open for various services and clients. DMZ forwarding does not interfere with established TCP or UDP connections, either to local ports or through configured or dynamic NAT rules. Outbound communications (private to public) from the DMZ server are handled in the same manner as the outbound communications from other hosts on that same private network.

WARNING! Security Warning: DMZ forwarding presents security risks for the DMZ server. Configure the DMZ forwarding option only if you understand and are willing to accept the risks associated with providing open access to this server and your private network.

dmzip=*ip address*

The IP address used for DMZ Forwarding.

maxentries=64-1024

The maximum number of concurrent NAT table entries that the device supports. This setting effectively limits the number of concurrent NAT rules and sessions that are permitted before disallowing them for resource constraint purposes. The maximum entries can range from **64** through **1024**. The default is **256**.

prenabled[1-4]={on|off}

Enables one of the four protocol-forwarding entries.

on

Enable this protocol-forwarding entry.

off

Disable this protocol-forwarding entry.

prnumber[1-4]={gre|esp}

The IP protocol whose packets will be forwarded for this entry.

gre

Indicates that the Generic Routing Encapsulation (GRE) protocol will be forwarded.

esp

Indicates that the Encapsulating Security Payload (ESP) protocol will be forwarded.

At this time, GRE and ESP (tunnel mode only) are the only protocols supported by the protocol-forwarding feature.

prtype[1-4]=*type*

This option is deprecated and unused by the device.

prip[1-4]=*ipaddress*

The IP address to which GRE packets will be forwarded.

poenabled[1-64]={on|off}
poproto[1-64]={tcp|udp|ftp}
pocount=[1-64]=*number of ports in range, minimum 1*
poexternal[1-64]=*number of ports in range, minimum 1*
pointernal[1-64]=*number of ports in range, minimum 1*
poip[1-64]=*ipaddress*

These **po*xxx*** options are grouped for each ***N*** in the [1-64] range to specify a single port forwarding rule. That is, the first port forwarding rule is defined by the values of: **poenabled1**, **poproto1**, **pocount1**, **poexternal1**, **pointernal1**, and **poip1**. The end of each option name specifies the index for the entry (**1-64**), for example, **poenabled1=on** or **poproto1=tcp**.

poenabled[1-64]={on|off}

Used to enable or disable one of the 64 port forwarding entries.

on

Enable this port forwarding entry.

off

Disable this port forwarding entry.

poproto[1-64]={tcp|udp|ftp}

The IP protocol associated with this port forwarding entry.

tcp

A TCP port is forwarded.

udp

A UDP port is forwarded.

ftp

The port forwarding rule is for an FTP server on the private side of the NAT. This keyword allows use of TCP ports other than TCP port **21** for private-side (or "inner") FTP servers.

pocount=[1-64]=*number of ports in range, minimum 1*

The number of consecutive ports in a port-forwarding range. This option allows you to forward more than one port in a single port-forwarding entry. When a range is configured, the first port in the range is specified, and the full range is indicated in the displayed entry information. The default is **1**. If the IP protocol for the port forwarding entry is FTP (**poproto[1-64]=ftp**), the value for this option can only be **1**.

poexternal[1-64]=*number of ports in range, minimum 1*

The external (or public) port that will be forwarded for this entry.

pointernal[1-64]=*number of ports in range, minimum 1*

The internal (or private) port to which packets will be forwarded for this entry. This value is a port number on the host whose IP address is specified by the **poip** option value for this entry.

poip[1-64]=*ipaddress*

The IP address of the host to which packets will be forwarded for this entry.

Examples

Enable NAT and specify settings for port forwarding entry 1

These example commands will enable NAT for the **mobile0** (cellular) interface as instance 1, with a maximum of 128 NAT entries (static or dynamic rules) permitted. A port forwarding rule is added to that NAT instance to enable the forwarding of TCP packets received at port **4009** of the public (mobile) interface of the device server, to TCP port **7008** of the host whose IP address is **143.191.1.228** on the Ethernet side of the device server.

```
#> set nat instance=1 ifname=mobile0 enabled=on maxentries=128
```

```
#> set nat instance=1 poenabled1=on poproto1=tcp poexternal1=4009
pointernal1=7008 poip1=143.191.1.228
```

In these examples, the **pocount1=1** option is not specified. The default is 1, unless otherwise specified or previously specified for the port forwarding rule entry in the indicated index slot.

Note that the forwarding to **143.191.1.228** through the Ethernet interface will occur only if the IP routing table (forwarding) of the device server is such that access to **143.191.1.228** is through that Ethernet interface. A typical NAT configuration has a public IP address on the WAN side and a private IP address and subnet on the LAN side. Sometimes, if the destination address for port forwarding isn't on the LAN subnet, then a static route is added to the IP routing table to make that forward as desired. The **set forwarding** command provides the means by which static routes can be configured.

Additional port forwarding rules can be configured for this enabled NAT instance=1 at any time, and they will be immediately made active in the device. Each rule should specify a different index as the last part of the **po*xxx*** options.

Display NAT and port/protocol forwarding settings

```
#> set nat
```

See also

- revert: The **revert nat** command reverts the settings configured by this command.
- set forwarding
- show vpn for information on NAT traversal (NAT-T).
- show: The **show nat** command shows the current NAT settings in a Digi device.
- Internet Engineering Task Force (IETF) document **IETF RFC 3715, "IPsec-Network Address Translation (NAT) Compatibility Requirements"** for information on NAT traversal.

set network

This should be updated to mention network port scan cloaking, its advantages, and commands for setting it.

Purpose

Sets and displays interface-specific settings for LAN or Wireless LAN (WLAN) networks, and global network settings that are independent of a LAN or WAN interface.

The interface-specific options are for Digi products that support multiple network interfaces. In Digi device firmware, the following interface names are in use:

- LAN interfaces: **eth0**, **eth1**, **wln0**. The interface-specific settings in this command apply to LAN interfaces only.
- WAN interfaces: **mobile0** (cellular), **wmx0** (WiMAX)
- The name **static** is for DNS server IP address values that can be configured using the **set network** options **dns1** and **dns2.** These values are not associated with a specific LAN or WAN interface.

The WAN interfaces (cellular or WiMAX) are configured/managed elsewhere in the command-line interface.

Global network settings include setting a gateway priority; that is, the default gateway is used to route IP packets to an outside network, unless controlled by another route, and the device's use of Domain Name Server (DNS). Additional options control the device's use of Transmission Control Protocol, and Address Resolution Protocol (ARP). The ARP options **garp** and **rto_min** can be used to optimize for latency at the network level.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-network=read** to display settings, and **set permissions s-network=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Set network configuration options

```
set network [interface specific options]
  [global options]
  [tcp keepalive options}
  [tcp time wait option]
  [tcp retransmit options]
  [arp options]
```

Where:

```
interface specific options:
  [interface=interface name]
  [ip=ip address]
  [submask=subnet mask]
  [gateway=gateway ip address]
```

```
  [static={on|off)]
  [dhcp={on|off}]
  [autoip={on|off}]
  [mtu=576-1500]

 Note: 'if' can be used as an abbreviation for 'interface'
```

```
global options:
 globalsettings
            valid names are: mobile0,eth0
  [dns1=primary dns server ip address]
  [dns2=secondary dns server ip address]
  [dnspriority=comma separated priority list]
             valid names are: static,mobile0,eth0
```

```
tcp keepalive options:
  [idle=10-86400] (seconds)
  [probe_count=5-30]
  [probe_interval=10-75] (seconds)
```

```
tcp retransmit options:
  [rto_min=30-1000] (milliseconds)
  [rto_max=1-240] (seconds)
```

```
tcp time wait option:
  timewait=(10-240) (seconds)
```

Note Changes to the TCP options do not affect existing connections.
TCP service listeners continue to use the previous option values until the service is restarted or a reboot is performed.

```
arp options:
  [arp_ttl=1-20] (minutes)
  [garp=30-3600] (seconds)
```

Display current network configuration options

```
set network
```

Options

interface specific options

Set configuration options for network interfaces for the Digi device.

interface=*interface name*

The name of the network interface. Any of the LAN interface names are valid for this option:

eth0

eth1

wln0

ip=*device ip address*

Sets the device IP address when DHCP is off. This option is only applicable if the **static** option is set to **on**.

submask=*subnet mask*

Sets the device submask address when DHCP is off. This option is only applicable if the **static** option is set to **on**.

gateway=*gateway ip address*

Sets the network gateway IP address.

The following three IP address options have a precedence. That is, if all three options are turned on, the order of precedence is: **static**, **dhcp**, **autoip**.

static={on|off}

When enabled, the device uses the specified IP address, gateway address, and submask. The default is **off**.

The **ip**, **submask**, and **gateway** options are meaningful only if **static=on** is configured. If **static=off**, IP configuration values that are applied to the device are obtained via DHCP or auto-IP. This is illustrated by the split display of the **set network** command output, in which one column shows values in use by the device, while the other shows the values configured in the settings. Even if **ip**, **submask**, and **gateway** are specified in the configuration settings, they are not actually used if **static=off**.

dhcp={on|off}

When enabled, the device attempts to use the DHCP protocol to find an IP address, gateway address, and submask. The default is **on**.

The **dhcp** option is enabled by default in almost all Digi devices, except **static** is enabled by default for these: all Digi Connect WAN products except Digi Connect WAN IA (which is DHCP default) and ConnectPort WAN VPN.

autoip={on|off}

When enabled, the device attempts to use the Auto IP protocol to find an IP address, gateway address, and submask. The default is **on**.

mtu=576-1500

This option allows for configuring a Maximum Transmit Unit (MTU) size octets (bytes) when sending packets through the specified network interface. This is the size of the IP packet that follows the network media-specific (for example, Ethernet) packet header. Setting this option limits the maximum size of outbound IP datagrams. Typically, and as the default value, the MTU is **1500** bytes.

A smaller MTU size than the default can be configured if necessary to communicate with remote hosts over a routed network connection that requires smaller packets and will not support IP fragmentation. While this

reduces the number of bytes per packet, which results in more packets being required to carry the same payload data, if a lot of data is sent, there are cases where a reduced maximum size is necessary.

For example, some routers in the Internet do not support maximum-size packets, particularly if they are being run through a tunnel (such as a VPN) in the path between sender and receiver. Although there is a convention/protocol called Path MTU Discovery (PMTUD) that can assist in such cases, using that protocol involves the use of ICMP packets to communicate back to the packet sender that it should send smaller packets. Some routers do not actually return those ICMP packets to the sender of the “too large” packet, which breaks the PMTUD mechanism. While this is more likely a problem for cellular and other WAN interfaces, it is the case that Ethernet and Wi-Fi also can be used as WAN interfaces in Digi products, such that they interface directly to the public Internet. In that case, setting the MTU size smaller of the **eth0** or **wln0** interface may be beneficial.

global options

globalsettings

Displays the global network settings.

gwpriority=*comma-separated interface name list*

A list of network interfaces, in priority order, used to determine the default gateway. The default gateway is used to route IP packets to an outside network, unless controlled by another route. All of the LAN and WAN interface names are valid for the **gwpriority** option value list:

eth0

eth1

wln0

mobile0

wmx0

A network interface can have a static gateway configured, or obtain a gateway from DHCP or other means when it is connected. The first interface in this list that supplies a gateway is used as the default gateway. The default gateway may change as interfaces connect and disconnect.

The IP Network Failover feature provides a dynamic method for selecting the default gateway. If IP Network Failover is properly configured and enabled, it overrides the gateway priority setting on this command. For a description of the failover feature and information on how to configure it, see "set failover" on page 219.

dns1=*primary dns server ip address*
dns2=*secondary dns server ip address*

For DNS, these options specify the DNS nameservers to use. Name lookups will be performed using the nameserver specified on **dns1** first, and if that fails, the nameserver specified on **dns2** will be used.

dnspriority=*comma separated priority list*

A list of DNS servers, in priority order, used to resolve computer host names. All of the LAN and WAN interface names, plus the keyword **static**, are valid for this value list:

eth0

eth1

wln0

mobile0

wmx0

static

Each type of server is tried, starting with the first in the list. For each server type, the primary server is tried first. If no response is received, the secondary server is tried. If neither server can be contacted, the next server type in the list is tried.

A network interface can obtain a DNS server from DHCP or other means when it is connected. If an interface does not obtain a DNS server, it is skipped and the next server in the priority list is tried.

tcp keepalive options

Options that configure how TCP keep-alive probes are sent.

The keep-alive options (**idle**, **probe_count**, **probe_interval**) should be configured for various services that are configured by **set service keepalive={on|off}**, or clients such as autoconnect, configured by **set autoconnect keepalive={on|off}**.

idle=10-86400

The amount of time, in seconds, to wait while not receiving TCP packets before sending out a keep-alive probe.

probe_count=5-30

The number of TCP keep-alive probes (specially formatted TCP frames) to send out before closing the TCP connection.

probe_interval=10-75

The amount of time, in seconds, to wait between sending TCP keep-alive probes.

tcp retransmit options

Options that control retransmission of TCP packets, including:

rto_min=30-1000 (milliseconds)

The lower bound or threshold for the TCP retransmission timeout (RTO), in milliseconds. The default is **1000** milliseconds.

TCP uses progressively larger retransmit values, starting at a minimum value that is calculated from a sliding window of ACK response round-trip times that is bounded at the bottom by **rto_min**. So, essentially, **rto_min** is not necessarily the timeout that will be used as the starting retransmit timeout, but it is the smallest such value that could be used.

This affects latency, because lowering **rto_min** ensures that retransmits take place in less time if they occur. By occurring sooner, the network is able to recover the lost data in less time at the expense of possibly retransmitting data that is still in-flight or successfully received by the other side, but unacknowledged due to a “delayed ACK” mechanism or something similar. Choosing a value lower than the default of **1000** milliseconds may help achieve improved latency performance when retransmissions occur.

rto_max=1-240 (seconds)

The upper bound or threshold for the TCP retransmission timeout (RTO), in seconds. When one side of a TCP connection sends a packet and does not receive an acknowledgment from the other side within the timeout period, the sending station retransmits the packet and sets an exponential backoff timeout. This is done for each successive retransmit until the maximum retransmission timeout is reached. Then, the TCP connection resets.

arp options

Options that control Address Resolution Protocol (ARP) requests.

arp_ttl=1-20 (minutes)

The initial value of the ARP time-to-live variable, which is the amount of time that an ARP entry remains in the network ARP cache. When an ARP cache entry first populated, the ARP time-to-live variable is set to this value. When the entry has existed in the table for this long without being updated, another ARP cache request is performed to make sure that there is not a new a new device at that IP.

garp=30-3600 (seconds)

The frequency of Gratuitous ARP (GARP) announcements. A Gratuitous ARP is a broadcast announcement to the network of a device's MAC address and the IP address being used for it. This allows the network to update its ARP cache tables without performing an ARP request on the network.

Gratuitous ARP announcements can affect latency in a limited way, because some systems stall or dispose of data that is transmitted during an ARP cache refresh. If this happens, setting the Gratuitous ARP frequency to be more often than the problem system's time-to-live variable can cause it to refresh the cache without needing to perform a request.

Examples

Configure global network settings

```
#> show network globalsettings
Global (interface independent) settings:

                      Currently in use by
                      the network stack          Stored configuration
                      --------------------       --------------------
  keepalive idle     : 7200                      7200
  probe count        : 9                         9
  probe interval     : 75                        75
  dns1               : 209.183.33.23             0.0.0.0
  dns2               : 209.183.33.23             0.0.0.0
  rto_min            : 1000                      1000
  rto_max            : 60                        60
  timewait           : 60                        60
  arp_ttl            : 15                        15
  garp               : 3600                      3600
  gwpriority         : mobile0,eth0
  dnspriority        : static,mobile0,eth0
```

Configure settings for Ethernet interface eth0

```
#> show network interface=eth0
"eth0" interface configuration:
  MAC Address        : 00:40:9D:36:DE:A8
                       Currently in use by
                       the network stack          Stored configuration
                       --------------------       --------------------
  ipaddress          : 10.30.1.121               192.168.1.1
  submask            : 255.255.255.0             255.255.255.0
  gateway            : 10.30.1.1                 0.0.0.0
  static             : off                       off
  dhcp               : supplied IP address       on
    dhcp server      : 10.30.1.11
    lease duration   : 86400 (seconds)
    renew after      : 43200 (seconds)
    rebind after     : 75600 (seconds)
```

```
    lease remaining  : 61943 (seconds)
  autoip             : on                          on
  mtu                : 1500                        1500
```

Manually set the device IP address

```
#> set network ip=10.0.0.1 gateway=255.255.255.0 submask=255.255.255.0
dhcp=off static=on autoip=off
```

Use DHCP to find an IP address, gateway address, and submask

```
#> set network static=off dhcp=on
```

Use DHCP or the Auto IP protocol to automatically configure network settings

```
#> set network static=off dhcp=on autoip=on
```

See also

- revert: The **revert network** command reverts the settings configured by this command.
- set autoconnect
- set dhcpserver
- set mobile and set mobileppp to configure settings for a cellular (mobile) network interface
- set sharing
- set service
- set wlan
- set wimax to configure settings for a WiMAX network interface.
- show: The **show network** command shows the current network settings in a Digi device.
- The “Latency tuning” chapter of the *Digi Connect Family User Guide* (Digi part number 9000565).

set orbcomm

Purpose

Changes the state of the built-in ORBCOMM® satellite modem driver to on or off.

The only instance the ORBCOMM modem driver should be turned off is when a Python-authored ORBCOMM serial application that can control a modem at a serial level is running on the Digi device. Disabling the functionality disables much of the behaviors of the other ORBCOMM commands.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-orbcomm=read** to display settings, and **set permissions s-orbcomm=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Change state of ORBCOMM satellite modem driver

```
set orbcomm state={on|off}
```

Display current state ORBCOMM satellite modem driver

```
set orbcomm
```

Options

state={on|off}

State of the ORBCOMM modem driver. The default state is **on**.

A reboot is required for the state change to take effect.

Example

```
#> set orbcomm

Operating System Management of ORBCOMM Satellite Modem

  state      : off

#> set orbcomm state=on
```

See also

- display orbcomm
- info orbcomm
- orbcomm

- revert: The **revert orbcomm** command reverts the settings configured by this command.
- set trace: The **orbcomm** option captures debugging information and error conditions from the ORBCOMM satellite modem subsystem.
- show: The **show orbcomm** command shows the current ORBCOMM settings in a Digi device.

set passthrough

Purpose

Configures the IP pass-through feature.

IP pass-through allows a Digi device to provide bridging functionality similar to a cable or DSL modem, where the Digi device device becomes “transparent” to the router or connected device. In this case, the router’s WAN interface believes it is connected directly to the mobile network, and has no knowledge that the Digi device is the mechanism providing that connectivity. The IP pass-through feature works with either cellular or WiMAX as the WAN interface.

A Digi device configured for IP pass-through, such as a ConnectPort WAN or Digi Connect WAN, passes its mobile IP address directly through and to the Ethernet device (router or PC) to which it is connected through the Ethernet port. From the perspective of the connected device, the Digi device essentially becomes transparent (similar to the behavior of a cable or DSL modem) to provide a bridge from the mobile network directly to the end device attached to the Digi device.

Since the mobile network address is effectively “passed-through” to the local device connected to the Ethernet port of the Digi device, all network access to it is bypassed, with some specific exceptions.

A reboot is required for IP pass-through settings to take effect.

Services disabled when IP pass-through is enabled

When you enable IP pass-through, the Digi device effectively disables all router and IP service functionality. Services that are disabled are:

- NAT
- Port Forwarding
- VPN
- Socket Tunnel
- Network Services configuration

Services available when IP pass-through is enabled

The Digi device is effectively transparent to all IP activity and network access by other devices, with these exceptions:

- It can be accessed via the serial port for configuration using the command line interface.
- It accepts TCP/IP connections for purposes of configuration by means of a “pinhole” on the mobile interface.
- It can be accessed by other devices on the local Ethernet segment via the default IP address of **192.168.1.1**.
- Clients such as SureLink, and client/server services such as remote manager client and server, are operational and enabled by default.

Using Pinholes to Manage the Digi Cellular Family Device

IP pass-through uses a concept called *pinholes*. You can configure the Digi device to listen on specific TCP ports, and terminate those connections at the Digi device for purposes of managing it. Those ports are called pinholes, and they are not passed on to the device connected to the Ethernet port of

the Digi device. Each pinhole command option specifies whether the network service and port are passed on to the device connected to the Ethernet port of the Digi device, or terminate at the Digi device. Network services or applications and ports that can be configured as pinholes include:

- Telnet network service: for accessing the device through a Telnet login and the command-line.
- SSH network service: for accessing to the device through a Secure Shell (SSH) login and the command-line.
- HTTP network service: for accessing the device through HTTP and the web interface.
- HTTPS network service: for accessing to the device through HTTPS and the web interface
- SNMP network service: for monitoring and managing the device through SNMP.
- Device Cloud remote management application (client-initiated connection)
- Digi SureLink application

For more information on the network services, see set service.

Device Cloud and Digi SureLink applications are automatically set up as pinholes so that they continue to work with the Digi device.

In addition, the Digi device uses a private address on the Ethernet interface strictly for use in configuration or local access. This allows a user on the local network to gain access to the web interface or a Telnet session to make configuration changes.

Remote Device Management and IP Pass-through

The Digi device allows you to enable pinholes for specific ports to allow remote users to manage the Digi device from the mobile network or open Internet. The Digi device retains its remote management capabilities using a Device Cloud server. The necessary pinholes are automatically defined when the Digi device is configured for IP Pass-through.This provides administrators with the same remote-management capabilities that exist in Digi remote devices.

Using the "set service" command with IP Pass-through

You can use the **set service** command to have a network service terminate both at a port on the Digi device and a different port on the connected device. For example, you could have the Digi device terminate the SSH service on port **2222**, and the connected device terminate SSH at port **22**. To do so, you would issue a **set service** command to move the SSH server from listening on port **22** to listening on port **222**. With such a configuration, both the Digi device and the connected box could respond to SSH.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-bridge=read"** to display settings, and **set permissions s-bridge=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure IP pass-through mode

```
set passthrough [state={enabled|disabled}]
  [proxyarp={enabled|disabled}]
  [subnetmask=subnet mask]
```

```
[http={pass|terminate}]
[https={pass|terminate}]
[telnet={pass|terminate}]
[ssh={pass|terminate}]
[snmp={pass|terminate}]
[dcloud={pass|terminate}]
[surelink={pass|terminate}]
[ping={pass|terminate}]
[ddnsupdate={pass|terminate}]
```

Display current IP pass-through mode settings

```
set passthrough
```

Options

state={enabled|disabled}

Enables or disables IP Pass-through.

proxyarp={enabled|disabled}

Enables or disables ARP proxy.

The existence of an entry in the proxy ARP table means that the Digi device responds to ARP requests for that IP address, as if the IP address were configured for the responding interface. This is generally useful in that the host making the ARP request forwards packets destined for that IP address to the Digi device, which will then forward them as the next routing hop.

For IP pass-through mode, if the “tethered” host connected to the LAN side of the Digi device is using DHCP to get its IP configuration from the Digi Device, it looks as if the Digi device is providing a subnet to that host even though the Digi device is only giving it the use of a single IP address (the WAN interface IP address). There is a possibility that the host must communicate with other hosts that appear to be in that subnet -- perhaps some other mobile device with an IP address in the same nearby address range. In that case, the Digi device must “proxy” the ARP requests that are received from the tethered host, so it sends them to the Digi to then forward to the WAN.

subnetmask=*subnet mask*

The IP address subnet.

http={pass|terminate}

Specifies whether the HTTP network service is configured to pass to the connected device or terminate at the Digi device for purposes of managing it, known as a pinhole.

https={pass|terminate}

Specifies whether the HTTPS network service is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**).

telnet={pass|terminate}

Specifies whether the Telnet network service is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**).

ssh={pass|terminate}

Specifies whether the SSH network service is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**).

snmp={pass|terminate}

Specifies whether the SNMP network service is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**).

dcloud={pass|terminate} ()

Device Cloud pass or terminate. Specifies whether the Device Cloud remote management application is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**). The default is **terminate**.

surelink={pass|terminate}

Specifies whether the SureLink application is configured to pass to the connected device (**pass**) or terminate at the Digi device for purposes of managing it, known as a pinhole (**terminate**). The default is **terminate**.

ping={pass|terminate}

Specifies whether ICMP echo (ping) requests pass to the connected device (**pass**) or terminate at the Digi device (**terminate**). The default is “pass.”

ddnsupdate={pass|terminate}

Specifies whether Dynamic DNS (DDNS) requests pass to the connected device (**pass**) or terminate at the Digi device (**terminate**). The default is **pass**.

See also

- display passthrough
- revert: The **revert passthrough** command reverts the settings configured by this command.
- show: The **show passthrough** command shows the current IP passthrough settings in a Digi device.
- The section on IP Pass-through settings in the *User Guide* for your Digi device.
- For descriptions of network services and their default network port numbers, see set service.
- For descriptions of the Device Cloud application and related settings, see set mgmtglobal and set mgmtnetwork.

set permissions

Purpose

Used to set user permissions associated with various services and command-line interface (CLI) commands, or display current permission settings.

Use of this command varies according to the user model implemented in the Digi device. For Digi products with one or two users, this command does not apply. It does apply to Digi products with two or more users. To determine the user model implemented in your Digi product, see User Models and User Permissions in Digi devices for more information.

Use of command options also depends on the features implemented in Digi devices. For example, **s-ethernet** is only supported in wired devices.

Some permissions keywords set for multiple commands, for example, the **s-ppp** permission keyword.

Permission descriptions

User permissions and their effects on commands are as follows.

Permission keyword	Affect on command execution
none	The command cannot be executed.
execute	The command can be executed.
r-self	The user can execute the **display** portions for both commands if the user is logged in on the specified line.
read	The user can execute the **display** portions for both commands for any line.
rw	The user can execute the **display** and **set** portions for both commands for any line.
rw-self	The user can execute the **display** and **set** portions for both commands if the user is logged in on the specified line.
w-self-r	The user can execute the **display** portions for both commands for any line and the **set** portions for both commands if the user is logged in on the specified line.

Commands without permissions

There are no permissions associated with the following commands:

- **close**
- **exit**
- **help**
- **quit**

Permissions for the "revert" command

For the **revert** command, the permissions associated with the various **set** commands are used, except for the **revert all** command variant, which uses a different mechanism that bypasses the individual **set** commands.

Permissions for the "show" commands

For the **show** command to display current device settings, the various **set** commands that configure the settings displayed by a "**show**" command must be set to either **read** or **r-self**, depending on the available permissions for the commands. "**show smscell** and **show vpn** have their own permissions, as stated in their command descriptions. See show smscell and show vpn.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the permissions settings for the line on which they are logged in: **set permissions s-permissions=r-self**.
- For a user to display the permissions settings for any line: **set permissions s-permissions=read**.
- For a user to display and set the permissions settings for the line on which they are logged in: **set permissions s-permissions=rw-self**.
- For a user to display the permissions settings for any line, and "**set group**" settings for the line on which the user is logged in: **set permissions s-permissions=w-self-r**.
- For a user to display and set the permissions settings on any line: **set permissions s-permissions=rw**. When permissions are set to **set permissions s-permissions=rw,** a user cannot set another user's permission level higher than their own level, and they cannot raise their own permission level.

Syntax

Set permissions

```
set permissions [type={user|group}]
  [{id=1-32|name={user name|group name}]
  [backup={none|execute}]
  [boot={none|execute}]
  [buffers={none|r-self|read|rw-self|w-self-r|rw}]
  [connect={none|execute}]
  [dhcpserver={none|execute}]
  [display={none|execute}]
  [filesys={none|read|rw}]
  [find-me={none|execute}]
  [fw-update={none|execute}]
  [iridium={none|execute}]
  [kill={none|execute}]
  [newpass={none|rw-self|rw}]
  [orbcomm={none|execute}]
  [ping={none|execute}]
  [python-cmd={none|execute}]
  [python-files={none|read|rw}]
  [reconnect={none|execute}]
  [revert-all={none|execute}]
  [rlogin={none|execute}]
  [s-accesscontrol={none|read|rw}]
  [s-alarm={none|read|rw}]
  [s-autoconnect={none|r-self|read|rw-self|w-self-r|rw}]
  [s-bridge={none|read|rw}]
  [s-camera={none|read|rw}]
  [s-cert={none|read|rw}]
  [s-dcloud-sms{none|read|rw}]
  [s-ddnsupdater=={none|read|rw}]
  [s-devicesecurity={none|read|rw}
  [s-dhcpserver={none|read|rw}]
  [s-dialserv={none|read|rw}]
  [s-dnsproxy={none|read|rw}]
  [s-ekahau={none|read|rw}]
  [s-ethernet={none|read|rw}]
  [s-gpio={none|read|rw}]
  [s-gps-geofence={none|read|rw}]
  [s-gps-static-position={none|read|rw}]
  [s-group={none|r-self|read|rw-self|w-self-r|rw}]
  [s-host={none|read|rw}]
  [s-hostlist={none|read|rw}]
  [s-ia={none|read|rw}]
  [s-login={none|read|rw}]
  [s-mesh={none|read|rw}]
  [s-mgmtconnection={none|read|rw}]
  [s-mgmtglobal={none|read|rw}]
  [s-mgmtnetwork={none|read|rw}]
  [s-net-failover={none|read|rw}]
  [s-network={none|read|rw}]
  [s-orbcomm={none|read|rw}]
  [s-permissions={none|r-self|read|rw-self|w-self-r|rw}]
  [s-pmodem={none|r-self|read|rw-self|w-self-r|rw}]
  [s-ppp={none|read|rw}]
```

```
[s-profile={none|r-self|read|rw-self|w-self-r|rw}]
[s-python={none|read|rw}]
[s-rciserial={none|r-self|read|rw-self|w-self-r|rw}]
[s-router={none|read|rw}]
[s-rtc={none|read|rw}]
[s-rtstoggle={none|r-self|read|rw-self|w-self-r|rw}]
[s-scan-cloak={none|read|rw}]
[s-serial={none|r-self|read|rw-self|w-self-r|rw}]
[s-service={none|read|rw}]
[s-sharing={none|read|rw}]
[s-sms-cellular={none|read|rw}]
[s-snmp={none|read|rw}]
[s-socket-tunnel={none|read|rw}]
[s-system={none|read|rw}]
[s-tcpserial={none|r-self|read|rw-self|w-self-r|rw}]
[s-term={none|read|rw}]
[s-time-source={none|read|rw}]
[s-trace={none|read|rw}]
[s-udpserial={none|r-self|read|rw-self|w-self-r|rw}]
[s-user={none|r-self|read|rw-self|w-self-r|rw}]
[s-vpn={none|read|rw}]
[s-vrrp={none|read|rw}]
[s-wimax={none|read|rw}]
[s-wlan={none|read|rw}]
[status={none|read|rw}]
[telnet={none|execute}]
[vpn={none|execute}]
[webui={none|execute}
[who={none|execute}]
```

Display current permission settings

```
set permissions
```

Options

type={user|group}

Specifies whether the command applies to users or groups. This option defaults to **user**.

{id=1-32|name={*user name*|*group name*}}

Identifies the user or group for which permissions are set.

id=1-32

The ID or the range of IDs of the users or groups to be acted on. If omitted, the “name” option must be specified.

name={*user name*|*group name*}

The name of the user or group to be acted on. If omitted, the **id** option must be specified.

backup={none|execute}

Permissions for the **backup** command. See backup .

boot={none|execute}

Permissions for the **boot** command. See boot.

buffers={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **display buffers** and **set buffer** commands. See display buffers and set buffer.

connect={none|execute}

Permissions for the **connect** command. See connect.

dhcpserver={none|execute}

Permissions for the **dhcpserver** command. See dhcpserver.

display={none|execute}

Sets permissions for:

- All the **display** commands. One permission setting applies to all **display** command variants.
- **flashdrv**. See flashdrv.
- **info ia**. See info ia.
- **info icmp**. See info icmp.

info ip. See info ip.

- **info iridium**. See info iridium.
- **info orbcomm.** See info orbcomm.
- **info serial**. See info serial.
- **info tcp**. See info tcp.
- **info time.** See info time.
- **info udp**. See info udp.
- **info wlan**. See info wlan.
- **info xbee**. See info xbee.

filesys={none|read|rw}

Permissions for user access to the Digi device's file system.

none

The user cannot access the file system.

read

The user can read the file system.

rw

The user can read and write the file system.

find-me={none|execute}

Permissions for the **findme** command. See findme.

fw-update={none|execute}

Permissions" for Digi device firmware update, performed either by the **boot load=*tftp_host*:*file*** command and through the web interface **Administration > Update Firmware** page. This permission covers updating of both EOS and POST firmware files.

iridium={none|execute}

Permissions for the **iridium** command.See iridium.

kill={none|execute}

Permissions for the **kill** command. See kill.

newpass={none|rw-self|rw}

Permissions for the **newpass** command. See "newpass" on page 140.

none

The command cannot be executed.

rw-self

The user can set their own password.

rw

The user can set any user's password.

orbcomm={none|execute}

Permissions for the **orbcomm** command. See orbcomm.

ping={none|execute}

Permissions for the **ping** command. See ping.

python-cmd={none|execute}

Controls the user's ability to directly run Python programs via the **python** command. See python. This permission is different from the one for executing Python programs via auto-start, which is configured by the **set python** command, and permissions controlled by the **s-python** permission.

python-files={none|read|rw}

Controls access to Python programs in the **Python** directory for the Digi device.

Note This option does not control access to Python programs by a user accessing the Digi device through Device Cloud. Instead, this keyword allows for visibility to the Python programs.

none

The user has no visibility of or access to the **Python** directory for the Digi device.

read

The user can view files in the **Python** directory for the Digi device.

rw

The user can view, add, or remove files in the **Python** directory for the Digi device.

reconnect={none|execute}

Permissions for the **reconnect** command. See reconnect.

revert-all={none|execute}

Permissions for the **revert all** command. See revert.

Individual **revert** commands are governed by the permissions for that particular command, but **revert all** uses a different mechanism that bypasses the individual commands.

rlogin={none|execute}

Permissions for the **rlogin** command. See rlogin.

s-accesscontrol={none|read|rw}

Permissions for the **set accesscontrol** command. See set accesscontrol.

s-alarm={none|read|rw}

Permissions for the **set alarm** command. See set alarm.

s-autoconnect={none|r-self|read|rw-self|w-self-r|rw}
Permissions for the **set autoconnect** command. See set autoconnect.
s-bridge={none|read|rw}
Permissions for the **set passthrough** command. See set passthrough.
s-camera={none|read|rw}
Permissions for the **set camera** command. See set camera.
s-cert={none|read|rw}
Permissions for the **certmgmt** command. See certmgmt.
s-dcloud-sms{none|read|rw}
Permissions for the **set dcloud_msgservice** command. See set dcloud_msgservice.
s-ddnsupdater={none|read|rw}
Permissions for the **set ddns** command. See set ddns.
s-devicesecurity={no ne|read|rw}
Permissions for the **set devicesecurity** command. See set devicesecurity.
s-dhcpserver={none|read|rw}
Permissions for the **set dhcpserver** command. See set dhcpserver.
s-dialserv={none|read|rw}
Permissions for the **set dialserv** command. See set dialserv.
s-dnsproxy={none|read|rw
Permissions for the **set dnsproxy** command. See set dnsproxy.
s-ekahau={none|read|rw}
Permissions for the **set ekahau** command. See set ekahau.
s-ethernet={none|read|rw}
Permissions for the **set ethernet** command. See set ethernet.
s-gpio={none|read|rw}
Permissions for the **set gpio** command. See set gpio.
s-gps-geofence={none|read|rw}]
Permissions for the **set geofence** command. See set geofence.
s-gps-static-position={none|read|rw}]
Permissions for the “set position” command. See set position.
s-group={none|r-self|read|rw-self|w-self-r|rw}
Permissions for the **set group** command. See set group.
s-host={none|read|rw}
Permissions for the **set host** command. See set host.
s-hostlist={none|read|rw}
Permissions for the **set hostlist** command. See set hostlist.
s-ia={none|read|rw}
Permissions for the **set ia** command. See set ia.
s-login={none|read|rw}
Permissions for the **set login** command. See set login.
s-mesh={none|read|rw}
Permissions for these commands

- **set xbee**. See set xbee.
- **show xbee.** See show.
- **revert xbee**. See revert.

s-mgmtconnection={none|read|rw}

Permissions for the **set mgmtconnection** command. See set mgmtconnection.

s-mgmtglobal={none|read|rw}

Permissions for the **set mgmtglobal** command. See set mgmtglobal.

s-mgmtnetwork={none|read|rw}

Permissions for the **set mgmtnetwork** command. See set mgmtnetwork .

s-net-failover={none|read|rw}

Permissions for the **set failover** command. See set failover.

s-network={none|read|rw}

Permissions for these commands:

- **set dirp**. See set dirp.
- **set network**. See set network.
- **set realport**. See set realport.

s-orbcomm={none|read|rw}

Permissions for the **set orbcomm** command. See set orbcomm.

s-permissions={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **set permissions** command itself.

s-pmodem={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **set pmodem** command. See set pmodem.

s-ppp={none|read|rw}

Permissions for these commands:

- **mobile_update**. See mobile_update.
- **provision**. See provision.
- **set mobile**. See set mobile.
- **set mobileppp**. See set mobileppp.
- **set ppp.** See set ppp.
- **set surelink**. See set surelink.

s-profile={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **set profile**" command. See set profile.

s-python={none|read|rw}

Permissions for executing the **set python** command. See set python.

To set permissions for executing Python programs via the **python** command, use the **python-cmd** permission.To control user visibility to the **Python** directory on the Digi device, use the **python-files** permission.

s-rciserial={none|read|rw}

Permissions for the **set rciserial** command. See set rciserial.

s-realport-usb={none|read|rw}

Permissions for these commands:

- **set realport**. See set realport.
- **set dirp.** See and set dirp.

s-router={none|read|rw}

Permissions for these commands

- **set forwarding**. See set forwarding.
- **set nat.**" See set nat.

s-rtc={none|read|rw}

Permissions for the **set time** command. See set time.

s-rtstoggle={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **set rtstoggle** command. See set rtstoggle.

s-scan-cloak={none|read|rw}

Permissions for the **set scancloak** command. See set scancloak.

s-serial={none|r-self|read|rw-self|w-self-r|rw}

Permissions for these commands:

- **set serial**. See set serial.
- **set switches**. See set switches.

s-service={none|read|rw}

Permissions for the **set service** command. See set service.

s-sharing={none|read|rw}

Permissions for these commands that control the port sharing feature:

- **revert sharing.** See revert.
- **set sharing**. See set sharing.
- **show sharing**. See show.

s-sms-cellular={none|read|rw}

Permissions for all commands associated with Short Message Service (SMS) support, including:

- **set smscell**. See set smscell.
- **show smscell**. See show smscell.
- **smscell**. See smscell.

s-snmp={none|read|rw}

Permissions for the **set snmp** command. See set snmp.

s-socket-tunnel={none|read|rw}

Permissions for the **set socket_tunnel** command. See set socket_tunnel.

s-system={none|read|rw}

Permissions for the **set system** command. See set system.

s-tcpserial={none|r-self|read|rw-self|w-self-r|rw}

Permissions for the **set tcpserial** command. See set tcpserial.

s-term={none|read|rw}

Permissions for the **set term** command. See set term.

s-time-source={none|read|rw}

Permissions for these commands:

- **set clocksource**. See "set clocksource" on page 190.
- **set time**. See set time.
- **set timemgmt**. See set timemgmt.

s-trace={none|read|rw}
Permissions for the **set trace** command. See set trace.
s-udpserial={none|r-self|read|rw-self|w-self-r|rw}
Permissions for the **set udpserial** command. See set udpserial.
s-user={none|r-self|read|rw-self|w-self-r|rw}
Permissions for the **set user** command. See set user.
s-vpn={none|read|rw}
Permissions for the **set vpn** command. See set vpn.
s-vrrp={none|read|rw}
Permissions for the **set vrrp** command. See set vrrp.
s-wlan={none|read|rw}
Permissions for the **set wlan** command. See set wlan.
status={none|read|rw}
Permissions for the **status** command. See status.
telnet={none|execute}
Permissions for these commands

- **newpass**. See newpass.
- **send**. See send.
- **telnet**. See telnet.

vpn={none|execute}
Permissions for the **vpn** command. See vpn.
webui={none|execute}
Permissions for access to the web interface for a Digi device.

none

The user cannot use the web interface.

execute

The user can access the web interface.

who={none|execute}
Permissions for the **who** command. See who.

Examples

Set user permissions

For user 1 defined in a Digi device, this command sets permissions for the **newpass**, **set user**, and **set group** commands to **read-write**:

```
#> set permissions id=1 newpass=rw s-user=rw s-group=rw
```

Set group permissions

For user group **gurus**, this command sets permissions for two commands: the **newpass** command is set to the **rw-self** permission and **set user** to the **read** permission.

```
#> set permissions type=group name=gurus newpass=rw-self s-user=read
```

See also

- User Models and User Permissions in Digi devices
- revert: The **revert auth** command reverts the settings configured by **set permissions**.
- set user
- set group
- show: The **show permissions** command shows the current permissions settings in a Digi device.

set pmodem

Purpose

Used to configure various options for modem emulation over TCP/IP, and display current modem-emulation settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the modem emulation settings for the line on which they are logged in: **set permissions s-pmodem=r-self**.
- For a user to display the modem emulation settings for any line: **set permissions s-pmodem=read**.
- For a user to display and set the modem emulation settings for the line on which they are logged in: **set permissions s-pmodem=rw-self**.
- For a user to display the modem emulation settings for any line, and set modem emulation settings for the line on which the user is logged in: **set permissions s-pmodem=w-self-r**.
- For a user to display and set the modem emulation settings on any line: **set permissions s-pmodem=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Configure modem emulation

```
set pmodem [port=range]
  [state={on|off}]
  [telnet={on|off}
  [ssl={on|off}]
  [auth={none|server|both}](used with “ssl=on” option only)
```

The connection-type options **telnet** and **ssl** apply to both incoming and outgoing calls via the pmodem feature. Only one of these options can be enabled at once, though all these options can be **off**.

Display modem-emulation settings

```
set pmodem [port=range]
```

Options

port=*range*

Used to specify the serial port. Optional on a single-port device.

state={on|off}

Used to enable or disable modem emulation on a given serial port.

on

Enables modem emulation.

off

Disables modem emulation.

The default is **off**.

telnet

Enables or disables Telnet processing on incoming and outgoing modem-emulation connections.

on

Enables Telnet processing.

off

Disables Telnet processing.

The default is **off**.

ssl={on|off}

Enables or disables SSL processing on incoming and outgoing modem-emulation connections

on

Enables SSL processing.

off

Disables SSL processing.

The default is **off**.

auth={none|server|both} (used with "ssl=on" option only)

Selects the authentication model when SSL is the enabled connection type (**ssl=on**).

none

No active verification of any peer's certificates. As a client, the Digi device presents a certificate if one is requested, but it will not validate a server certificate. As a server, the Digi device present its certificate, but does not request or validate a client certificate.

server

The Digi device actively verifies only server certificates. As a client, the Digi device verifies the server certificate if one is presented to it, and presents a certificate if one is requested. As a server, the Digi device presents its certificate, but does not request or validate a client certificate.

both

The Digi device actively verifies client and server certificates. As a client, the Digi device verifies the server certificate, and presents a certificate if one is requested. As a server, the Digi device presents its certificate, requests a client certificate, then validates the certificate.

Example

```
#> set pmodem port=1 state=on
```

See also

- certmgmt for additional information on managing certificates.
- revert: The **revert pmodem** command reverts the settings configured by this command.
- set network
- set profile
- show: The **show pmodem** command shows the current modem emulation settings in a Digi device.
- Chapter 3, Modem Emulation AT Command Set for descriptions of Digi-specific commands for modem-emulation configurations.

set position

Purpose

Defines a position for static Digi device. This allows an end user to keep track of the geographic location of various devices. The position parameters can be queried with the Remote Command Interface (RCI) protocol, and this information can be used by applications such as Device Manager.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-gps-static-position=read** to display settings, and **set permissions s-gps-static-position=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set position [state={enabled|disabled}]
  [latitude={-90.000000-90.000000}]
  [longitude={-180.000000-180.000000}]
```

Options

state={enabled|disabled}

Enables or disables the position for a static Digi device.

latitude={-90.000000-90.000000}

Defines the latitude component of the Digi device, in degrees.

longitude={-180.000000-180.000000}

Defines the longitude component of the Digi device, in degrees.

Example

```
#> set position state=enabled latitude=63.1833333 longitude=14.65
```

See also

- display gps
- display provisioning displays the position of the static Digi device.
- revert: The **revert position** command reverts the settings configured by this command.
- set geofence
- show. The **show position** command shows the current position settings in a Digi device.
- The *Remote Command Interface (RCI) Specification*, Digi part number 90000569
- GPS sample application code in *Digi ESP for Python*. Use the Python sample wizard to locate the samples.

set ppp

Purpose

Configures or displays Point-to-Point Protocol (PPP) outbound connections.

This command can also be used to enable or disable all mobile connections.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display settings, and **set permissions s-ppp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure PPP connections

```
set ppp port=range
  [state={enabled|disabled}]
  [auth_method={none|PAP|CHAP|both}]
  [passive={on|off}]
  [remote_address={ip address|negotiated}]
  [local_address={ip address|negotiated}]
  [address_mask=ip address mask]
  [default_gateway={yes|no}]
  [ipcp_dns_enabled={on|off}]
  [protocol_compression={on|off}]
  [address_compression={on|off}]
  [header_compression={on|off}]
  [lcp_keepalive={on|off}]
  [lcp_ka_quiet_time=(10-86400 seconds)]
  [lcp_ka_max_missed_replies={(2-255|0=ignore missed replies)}]
  [asyncmap=hex string]
  [chap_id=chap id]
  [chap_key=chap key]
  [pap_id=pap_id]
  [pap_password=pap password]
  [mru=1500-2048
  [mtu=1500-2048]
  [redial_attempts=attempts]
  [redial_delay=delay]
  [rx_idle_timeout=timeout]
  [tx_idle_timeout=timeout]
  [init_script=chat script]
  [dial_script=chat script]
  [login_script=chat script]
  [n{1-4}=phone_number]
  [proxy_arp={on|off}
  [device_description="description text"]
```

Display PPP settings

```
set ppp
```

Enable or disable all mobile connections

Enter this command with no other options specified:

```
set ppp port=5 state={enabled|disabled}
```

Options

port=*range*

The physical interface to which the PPP outbound configuration applies. Required.

state={enabled|disabled}

The state of the interface. The default is **disabled**.

auth_method={none|PAP|CHAP|both}

Determines whether authentication is required for outbound PPP connections and, if so, what kind.

none

The remote user does not require PPP authentication.

pap

Password Authentication Protocol (PAP) authentication is required.

chap

Challenge Handshake Authentication Protocol (CHAP) authentication is required. CHAP authentication works between two Digi devices. CHAP will be negotiated to PAP for all other connections.

both

Both CHAP and PAP authentication are required.

The default is **none**.

passive={on|off}

Specifies whether the device server waits for the remote system to begin PPP negotiations, or can initiate PPP negotiations on its own.

Do not set both sides of a PPP connection to **passive=on**.

on

The device server waits for the remote system to begin PPP negotiations.

off

The device server may initiate PPP negotiations.

The default is **off**.

remote_address={*ip address*|negotiated}

The address of the peer at the other end of the outbound PPP connection. Either a specific address or the keyword **negotiated** can be specified; **negotiated** means that the address will be accepted from the peer. An IP address of all zeroes is equivalent to specifying the keyword **negotiated**.

local_address=*ip address*|negotiated}

The IP address of the local end of the PPP outbound connection. Either a specific address or the keyword **negotiated** can be specified; **negotiated** means that the address will be accepted from the peer. An IP address of all zeroes is equivalent to specifying the keyword **negotiated**.

address_mask=*ip address mask*

The IP mask to apply to the address specified on the **remote address** and **local address** options. When you specify a specific IP address on the **remote address** and **local address** options, this option modifies the meaning of the IP address for routing purposes. The default is **255.255.255.255**.

default_gateway={yes|no}

Selects whether to use the PPP interface as the default route. The default is **no**.

ipcp_dns_enabled={on|off}

Enables or disables the IPCP (PPP Internet Protocol Control Protocol) acquisition of DNS IP addresses. This option is enabled by default to preserve prior behavior.

protocol_compression={on|off}

Specifies whether the device server attempts to negotiate protocol compression on PPP connections.

on

The device server attempts to negotiate protocol compression on PPP connections.

off

The device server will **not** negotiate protocol compression.

The default is **on**.

address_compression={on|off}

Specifies whether the device server attempts to negotiate address compression on PPP connections.

on

The device server attempts to negotiate address compression.

off

The device server does **not** attempt to negotiate address compression.

The default is **on**.

header_compression={on|off}

Specifies whether the device server attempts to negotiate IP protocol header compression on PPP connections. This is commonly referred to as Van Jacobsen (VJ) header compression.

on

The device server attempts to negotiate IP protocol header compression.

off

The device server does not attempt to negotiate IP protocol header compression.

The default is **on**.

lcp_keepalive={on|off}

Specifies whether the device server sends Link Control Protocol (LCP) echo requests after a "quiet" interval, in order to test the PPP link and/or keep it alive. "Quiet" means not having received any bytes over the PPP link for a specified time interval, which is set by the **lcp_ka_quiet_time** option. In PPP networks that support LCP echoes, an LCP echo reply is returned by the remote end of the PPP connection.

Even if LCP keepalives are disabled in this device (by **lcp_keepalive=off**), the device will still reply to LCP echo request messages it may receive from the remote side of the PPP connection by sending an LCP echo reply message. But the device itself will not originate any LCP echo request messages.

The options are:

on

The device server sends LCP echo requests after a configurable "quiet" interval, set by the **lcp_ka_quiet_time** option.

off

The device server does not send LCP echo requests.

lcp_ka_quiet_time=10-86400 seconds

Specifies the "quiet" interval, in seconds, after which the device server sends an LCP echo request. "Quiet" means not having received any bytes over the PPP link for the interval specified by this option.

lcp_ka_max_missed_replies={2-255|0=ignore missed replies}

Specifies how many consecutive echo replies may be missed before the device server disconnects the PPP link. A value of **0** (zero) specifies that the device server should not act on missed LCP echo replies by disconnecting the PPP link. Note that if bytes of any kind, LCP echo reply or otherwise, are received, the PPP link is deemed to be active, and the "missed LCP echo replies" count is reset to **0**.

asyncmap=*hex string*

A mask for PPP connections that defines which of the 32 asynchronous control characters to transpose. These characters, in the range **0x00** to **0x1f**, are used by some devices to implement software flow control. These devices may misinterpret PPP transmission of control characters and close the link. This mask tells PPP which characters to transpose.

The default is **FFFF**, which means transpose all 32 control characters. Any combination is valid. The following are the masks most likely used:

FFFFFFFF

Transpose all control characters.

00000000

Transpose none.

000A0000

Transpose **Ctrl-Q** and **Ctrl-S**.

chap_id=*chap id*

A character string that identifies the outbound PPP user using CHAP authentication. This is equivalent to a user or login name. The string must be **32 or fewer** characters and must be recognized by the peer.

chap_key=*chap key*

A character string that authenticates the outbound PPP user using CHAP authentication. This is equivalent to a password. The string must be **16 or fewer** characters and must be recognized by the peer.

pap_id=*pap id*

A character string that identifies the outbound PPP user using PAP authentication. This is equivalent to a user (or login) name. The string must be **32 or fewer** characters and must be recognized by the peer.

pap_password=*pap password*

A character string that authenticates the outbound PPP user using PAP authentication. This is equivalent to a password. The string must be **16 or fewer** characters and must be recognized by the peer.

mru=1500-2048

The maximum received unit (MRU), or frame size, in bytes, to be received from the other end of the PPP connection. This is a negotiated value. The default is **1500** bytes.

mtu=1500-2048

The maximum transmission unit (MTU), or frame size, in bytes, to use for this PPP outbound connection.

For PPP connections, the MTU is negotiated. Therefore, enter **1500**, the largest size the Digi device will permit the remote host to send.

For PPP users, the range is **128** to **1500** bytes, and the default is **1500** bytes.

redial_attempts=*attempts*

The number of times the firmware will attempt to redial before giving up.

redial_delay=*delay*

The time to wait after an unsuccessful dial attempt.

rx_idle_timeout=*timeout*

The time, in seconds, after which if no data has been received over the link, the PPP connection is disconnected.

tx_idle_timeout=*timeout*

The time, in seconds, after which if no data has been transmitted over the link, the PPP connection is disconnected.

init_script=*chat script*

An initialization script, run once at interface startup. For example:

```
init_script="'' ATZ OK \c"
```

dial_script=*chat script*

A dialing script, used any time a number is dialed for the interface. For example:

```
dial_script="'' ATDT\T CONNECT \c"
```

login_script=*chat script*

A login script, used to log in to the remote system on the other end of the outbound PPP connection. For example:

```
login_script="ogin: <username> assword: <password>"
```

n{1-4}=*phone_number*

Up to four phone numbers to dial to request a PPP outbound connection. The phone numbers are dialed sequentially.

proxy_arp={on|off}

When enabled, performs proxy ARP for the remote peer of the PPP session, so that the peer can be made to appear on our local network. It performs that proxy ARP on the subnet to which the IP address assigned to that peer belongs.

device_description="*description text*"

An alternate string used for SNMP purposes. If not specified in the settings, the PPP device name (for example, **ppp4**) is used instead. This string allows for SNMP-produced information to fit better with network management software that prefers identification strings to be in a format such as **mobile0**.

See also

- display pppstats: The **display pppstats** command displays the current status of PPP connections. See PPP status and activity Information returned by display pppstats for descriptions of the status information.
- revert: The **revert ppp** command reverts the settings configured by this command.
- show: The **show ppp** command shows the current PPP outbound connection settings in a Digi device.

set profile

Purpose

Associates a particular port with one of several port configuration profiles, or displays the current port-profile settings.

Port profiles are a defined set of port configuration settings for a particular use. A port profile reconfigures serial-port settings to the necessary default values in order for the profile to operate correctly.

Port-profile configuration is most often performed through the web interface for a device. It is not often specified from the command line, but is available if needed.

Digi devices support several port profiles, shown in the command syntax. The profiles supported on your Digi device may vary.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the profile settings for the line on which they are logged in: **set permissions s-profile=r-self**.
- For a user to display the profile settings for any line: **set permissions s-profile=read**.
- For a user to display and set the profile settings for the line on which they are logged in: **set permissions s-profile=rw-self**.
- For a user to display the profile settings for any line, and set modem emulation settings for the line on which the user is logged in: **set permissions s-profile=w-self-r**.
- For a user to display and set the profile settings on any line: **set permissions s-profile=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Configure port profile settings

```
set profile port=port
  profile={unassigned|console_management|local_config|
  modem_emulation|realport|tcp_sockets|tunneling|udp_sockets|
  custom|ia|dialserv|gps}
```

Display current port profile settings for all available serial ports

```
set profile
```

Display current port profile settings for a particular serial port

```
set profile port=port
```

Options

port=*port*

The serial port number or range of serial ports associated with the port profile. Required when configuring port profiles.

profile={unassigned|console_management|local_config|modem_emulation|realport|tcp_sockets|tunneling|udp_sockets|custom|ia|dialserv}

The port profile to use for the serial port. Required when configuring port profiles. Choosing a particular port profile causes the serial port's configuration to be reset to defaults, and then for the default settings for that port profile to take effect.

Depending on the port-profile choices available for the device, the value of **profile** can be one of the following:

unassigned

No port profile is assigned to the port. This option can be used to clear a previously assigned profile and its associated serial settings from the port.

console_management

Associates the **Console Management** port profile with the port. This profile allows access to a device's console port over a network connection.

local_config

Associates the **Local Configuration** port profile with the port.This profile configures the serial port to act as a modem.

modem_emulation

Associates the **Modem Emulation** port profile with the port. This profile allows connections to standard terminals or terminal emulation programs to the serial port, to use the serial port as a console to access the command line interface.

realport

Associates the **RealPort** port profile with the port. This profile allows mapping a COM or TTY port to the serial port.

tcp_sockets

Associates the **TCP Sockets** port profile with the port. This profile allows a serial device to communicate over a TCP network.

When the **TCP Sockets** profile is set, the DTR flow-control signal indicates when a TCP socket connection has been established. This information can be useful in monitoring the serial line and using it as a flow-control mechanism to determine when the Digi device is connected to a remote device with which communication is being established. This mechanism can be combined with using the **DCD** signal to close the connection and the **DSR** signal to do RCI over serial. Together, these signals can be used to make the Digi device auto connect to many devices, deterministically, on the network.

For ConnectPort X2 gateways, **TCP Sockets** can be used to directly access the XBee RF module on a gateway. See Direct Access communication with the XBee RF module on ConnectPort X2 gateways.

tunneling

Associates the **Serial Bridge** port profile with the port. This profile is known in the web interface as the Serial Bridge profile. It configures one side of a serial bridge. A serial bridge connects two serial devices over the network, as if they were connected with a serial cable.

udp_sockets

Associates the **UDP Sockets** port profile with the port. This profile allows a serial device to communicate using UDP.

custom

Associates the **Custom** port profile with the port.This profile is an advanced option to allow full configuration of the serial port. It allows you to view all settings associated with the serial port.

ia

Associates the **Industrial Automation (IA)** port profile with the port. IA profile, which configures the serial port for use in Industrial Automation (IA). The default configuration settings assume Modbus/RTU slaves with addresses **1** to **32** are attached to the serial port. Default port characteristics are **9600:8,N,1**. Unit ID zero (0) is auto-mapped to Modbus/RTU slave address 1. The electrical interface is set as EIA-232, EIA-422, or EIA-485 by the four DIP switches on the bottom of the unit.

dialserv

Associates the **Dialserv** port profile with the port. Digi DialServ is an RJ-11 phone line simulator that allows legacy devices with built-in modems to communicate across LANs/WANs. This profile configures the Digi device to connect/tunnel serial data to an external host when the DialServ receives an incoming call, causes the DialServ to make outgoing calls, and tunnels TCP data from the incoming connection over the Dialserv when TCP traffic is received on the configured ports on the Digi device.

Important: Use of this profile is **required** for DialServ interoperation.

gps

Associates the **GPS** port profile with the port. This profile allows the Digi device to make use of an NMEA-0183 compliant Global Positioning System (GPS) data stream for location and geofencing.

Example

```
#> set profile port=1 profile=realport
```

See also

- display serial
- info serial
- revert: The **revert profile** command reverts the settings configured by this command.
- set dialserv for a description of configuring behaviors on serial ports with DialServ devices.
- set geofence and set position for configuring a Digi device for use with a Global Positioning System (GPS) and geofencing application.
- set ia for a description of the default settings for Industrial Automation.
- set pmodem
- set realport
- set serial
- set tcpserial
- set udpserial
- show: The **show profile** command shows the current port profile settings in a Digi device.

- For more information on port profiles, see the topic **About Port Profiles** in your Digi product's *User Guide*.

set putty

Purpose

Configures terminal-emulation settings for ConnectPort Display, and displays current terminal-emulation settings.

ConnectPort Display can emulate a terminal connected to a host/server over a serial line or the network. When connected over the network RealPort must be installed on the server. RealPort ports appear to applications on the server as serial ports, but the data is redirected over the network to the terminal. For more information on RealPort, see the *RealPort Installation Guide*.

A ConnectPort Display device can emulate a terminal connected to a host/server. Data sent from the host application is processed and displayed on the terminal screen. A keyboard can also be used. If a keyboard is connected to the terminal, the terminal data is sent to the host application for it to process.

A reboot is required for the terminal-emulation settings to take effect.

Syntax

Set general terminal emulator options

```
set putty [state={on|off}]
  [width={80|132}]
  [height=10-60]
  [hostport={/com/0|/com/1|/vcom/0]
  [keyboardport={/com/0|/com/1}]
  [cursortype={none|block|underline|vertical}]
  [blinkcursor={on|off}]
  [blinktext={on|off}]
  [backspaceisdelete={on|off}]
  [lfimpliescr={on|off}]
  [characterset=host charset]
```

Set key mappings - range required

```
set putty
[deletekeymaprange=1-32]
[keymaprange=1-32]
[inseq=00-FF]
[outseq=00-FF]
```

Display terminal emulation settings

```
set putty
```

Options

General terminal emulation options

state={on|off}

Enables or disables the terminal emulator.

width={80|132}

The default width of the terminal, specified as the number of columns of text to display on the terminal emulator. The default width is **80**.

height=10-60

The default height of the terminal, specified as the number of rows of text to display on the terminal emulator. The default height is **24**.

hostport={/com/0|/com/1|/vcom/0}

Specifies how the terminal emulator connects to a host application, and how it reads input from the host. The terminal emulator reads input from a host application and displays it on the screen. Input can be read over one of the serial ports on the ConnectPort Display, or over the network. Network connections are achieved using Realport.

Valid values are **/com/0** and **/com/1** (serial ports **1** and **2**) and **/vcom/0** (network via RealPort). The default is **/com/0**.

When using a network connection, you must install the RealPort driver on the host. This will create a virtual COM port for each serial port on your ConnectPort Display (these are the traditional RealPort COM ports) as well as one additional virtual COM port that can be used for the terminal emulator connection. The host application must be configured to use this additional virtual COM port.

keyboardport={/com/0|/com/1|No Keyboard}

Specifies how a keyboard, if used, is connected to the terminal emulator. Connecting a keyboard is optional. The terminal emulator can read keyboard input from one of the serial ports. Keyboard data is then passed back up to the host application over the host connection.

Valid values are **/com/0** and **/com/1** (serial ports **1** and **2**) and **No Keyboard**. The default is **/com/1**.

In some environments, the keyboard data should not be passed back up to the host application over the host connection. In this case, you can still connect a keyboard to a serial port, and simply treat it like any other serially connected device. To do so, you would configure the terminal emulator to use **No Keyboard** for the keyboard connection, and then configure the serial port for the keyboard to use the RealPort port profile. Keyboard data would then be sent to the host system over the standard RealPort COM port. In this case, the host application reads keyboard data from one COM port and writes host data to a different COM port.

cursortype={none|block|underline|vertical}

Specifies how the cursor appears on the terminal emulator display: as a block, an underline, a vertical line, or no cursor.

none

The cursor has no visible display characteristics.

block

The cursor is displayed as a block.

underline

The cursor is displayed as an underline (underscore) character.

vertical

The cursor is displayed as a vertical bar.

The default is **underline**.

blinkcursor={on|off}

Enables or disables blinking of the cursor. The default is **on**.

blinktext={on|off}

Enables or disables the use of blinking text.The terminal emulator can display text that blinks on and off. This setting allows you to turn off blinking text. When blinking text is disabled and the terminal emulator attempts to make some text blink, the text will instead be displayed with a bold background color. The default is **on**.

backspaceisdelete={on|off}

This option allows you to choose which code, ASCII code **8** or **127**, is generated and sent to the host when the **Backspace** key is pressed. On some terminals, pressing the **Backspace** key sends the same code as **Ctrl-H** (ASCII code **8**). On other terminals, pressing the **Backspace** key sends ASCII code **127** (usually known as **Ctrl-?** or **Delete**), so that the action can be distinguished from **Ctrl-H**. The default is **on**.

lfimpliescr={on|off}

Specifies whether an **LF** (Line Feed) character includes an implicit **CR** (Carriage Return) character.

Most servers send two control characters, **CR** and **LF**, to start a new line of the screen. The **CR** character makes the cursor return to the beginning of the current line of text. The **LF** character makes the cursor move one line down. Some servers only send **LF**, and expect the terminal to move the cursor over to the left automatically. If your server does this, you will see a stepped effect on the screen. If this happens, try enabling this setting.The default is **off**.

characterset=*host charset*

The character set for data received from the host. During a session, the terminal emulator receives a stream of 8-bit bytes from the server, and in order to display them on the screen it needs to know the character set in which to interpret these streams of bytes.

There are several character sets from which to choose. A few notable character sets are:

- The **ISO-8859** series are all standard character sets that include various accented characters appropriate for different sets of languages.
- The **Win125x** series are defined by Microsoft for similar purposes. Win1252 is almost equivalent to ISO-8859-1, but contains a few extra characters such as matched quotes and the Euro symbol.
- **CP437** contains the old IBM PC character set with block graphics and line-drawing characters. This is also used on MS-DOS systems.
- **UTF-8** contains unicode data interpreted as being in the UTF-8 encoding. Not all server applications will support UTF-8.

The default is **ISO-8859-1**.

The complete list of allowed character sets is:

Character Set name	Description
ISO-8859-1	ISO-8859-1:1998 (Latin-1, West Europe)
ISO-8859-2	ISO-8859-2:1999 (Latin-2, East Europe)
ISO-8859-3	ISO-8859-3:1999 (Latin-3, South Europe)
ISO-8859-4	ISO-8859-4:1998 (Latin-4, North Europe)

Character Set name	Description
ISO-8859-5	ISO-8859-5:1999 (Latin/Cyrillic)
ISO-8859-6	ISO-8859-6:1999 (Latin/Arabic)
ISO-8859-7	ISO-8859-7:1987 (Latin/Greek)
ISO-8859-8	ISO-8859-8:1999 (Latin/Hebrew)
ISO-8859-9	ISO-8859-9:1999 (Latin-5, Turkish)
ISO-8859-10	ISO-8859-10:1998 (Latin-6, Nordic)
ISO-8859-11	ISO-8859-11:2001 (Latin/Thai)
ISO-8859-13	ISO-8859-13:1998 (Latin-7, Baltic)
ISO-8859-14	ISO-8859-14:1998 (Latin-8, Celtic)
ISO-8859-15	ISO-8859-15:1999 (Latin-9, “euro”)
ISO-8859-16	ISO-8859-16:2001 (Latin-10, Balkan)
CP437	CP437 (IBM-437/MS-DOS Latin, United States)
CP850	CP850 (IBM-850/MS-DOS Latin 1, West Europe)
CP1250	Win1250 (Central European)
CP1251	Win1251 (Cyrillic)
CP1252	Win1252 (Western)
CP1253	Win1253 (Greek)
CP1254	Win1254 (Turkish)
CP1255	Win1255 (Hebrew)
CP1256	Win1256 (Arabic)
CP1257	Win1257 (Baltic)
CP1258	Win1258 (Vietnamese)
KOI8-R	
KOI8-U	
Mac Roman	
Mac Turkish	
Mac Croatian	
Mac Iceland	
Mac Romanian	
Mac Greek	

Character Set name	Description
Mac Cyrillic	
Mac Thai	
Mac Centeuro	
Mac Symbol	
Mac Dingbats	
Mac Ukraine	
Mac VT100	
VISCII	
HP ROMAN8	
DEC MCS	
UTF-8	

Key mapping terminal emulation options

Character codes received from a keyboard can be converted to different character codes before being sent to the host. This conversion, known as key mapping, can be useful when you have different types of keyboards that need to be mapped to the same set of character codes.

A key mapping consists of an input sequence of character codes and the output sequence of codes to which they will be converted. Generally, you would specify both the input and output sequences as single character codes, although you can define up to 5 character codes for each. A character code is entered as two hexadecimal digits. For example:

- To convert the ASCII character **A** to **B**, you would define the input and output sequences as **41** and **42** respectively, which are the hexadecimal representations of the ASCII characters.
- To convert a code of decimal **10** to **0**, you would define the input and output sequences as **0A** and **00**, respectively.

Note that character codes are always two hexadecimal digits, which means that leading zeroes must be provided.

A key mapping entry requires a range, specified by **keymaprange**, and at least an input sequence, specified by **inseq**. The output sequence (**outseq**) is optional. When removing a key mapping entry, only **deletekeymaprange** is required.

The keymap entries are held in a table, as are other device settings such as UDP, serial destinations, and alarms. When adding a new entry (an **inseq**/**outseq** pair), you specify at what index in the table to add it using **keymaprange**.To delete an entry (or range of multiple entries) you specify the index/range with **deletekeymaprange**. Note that the **Terminal Emulation** settings in the web interface manages the indexes for you. If you do not want to deal with the key mappings at an index level, you can configure the key mapping through that interface.

Options specified for key mapping include:

deletekeymaprange=1-32

Removes the key mapping entry at the specified index or range of indexes.

keymaprange=1-32

The index/range used when adding new key mapping entries or replacing existing ones.

inseq=00-FF

The input key sequence, specified as two hexadecimal digits.

outseq=00-FF

The output sequence, specified as two hexadecimal digits.

Example

Display current terminal emulation settings

```
#> show putty
Terminal Configuration :

state = on
width = 80
height = 24
hostport = /com/0
keyboardport = /com/1
cursortype = underline
blinkcursor = on
blinktext = on
backspaceisdelete = on
lfimpliescr = off
characterset = ISO-8859-1
Key Map: range                  inseq      outseq
       1                                      A1           F1
       2                                      A2           F2
       3                                      A3           F3
```

Configure general terminal emulation settings

Given the above settings, to adjust the screen height and cursor type, you would enter:

```
# set putty height=30 cursortype=vertical
```

Add, replace, and delete entries in the key mapping table

To add/replace the first 3 entries in the table you would use the following commands:

```
#> set putty keymaprange=1 indeq=A1 outseq=F1
```

```
#> set putty keymaprange=2 indeq=A2 outseq=F2
```

```
#> set putty keymaprange=3 indeq=A3 outseq=F3
```

Now, to delete the first 2 entries:

```
#> set putty deletekeymaprange=1-2
```

You are left with one keymap entry, and it is at index 3, so to delete this last one enter:

```
#> set putty deletekeymaprange=3
```

See also

- revert: The **revert putty** command reverts the settings configured by this command.
- set serial
- set video
- show: The **show putty** command shows the current terminal emulation settings in a Digi device .
- The *ConnectPort Display User Guide* section on configuring terminal emulation settings.

set python

Purpose

Configures Python programs to execute when the Digi device boots.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-python=read** to display settings, and **set permissions s-python** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set python [range=1-4]
  [state={on|off}]
  [onexit={none|restart|reboot}]
  command="program file [arguments...]"
```

Options

range=1 – 4

The index or indices to view or modify with the command.

state={on|off}

Enables or disables Python program autostart, or automatic execution of Python programs loaded on the Digi device when the device boots. When the state is set to **on**, the command specified runs when the device boots.

onexit={none|restart|reboot}

The action that should occur if the specified Python program exits.

none

No action.

restart

Restart the specified Python program.

reboot

Reboot the Digi device.

command="*program file* [*arguments*...]"

The program filename to execute, including any arguments to pass with the program, similar to the arguments for the **python** command. While this option allows for programs to run from a TFTP server, this use is not recommended. If there are spaces to provide arguments, make sure to wrap the entire command in quotation marks.

Examples

```
#> set python range=1 state=on onexit=restart
```

See also

- python to execute Python programs.
- revert: The **revert python** command reverts the settings configured by this command.
- who to view which Python threads are running. Python threads are listed for informational purposes. They cannot be killed with either the **kill** command or via the **Management > Connections** page in the web interface.
- The *Digi Python Programming Guide* to learn more about the Python programming language as implemented in Digi products, and writing Python programs. This guide is available on the Digi website at:
  www.digi.com/wiki/developer/index.php/Digi_Python_Programmer's_Guide
- The Digi Developer Community Wiki is a place to learn about developing solutions using Digi's communications portfolio, software and services, including Python, Device Cloud, DIA, and more.

  Digi's Developer Wiki is where you will learn about developing solutions using Digi's communications product, software and services. The Wiki includes how-to information, example code, and M2M information to speed application development. Digi encourages an active developer community and welcomes your contributions.

  www.digi.com/wiki/developer/index.php/Main_Page
- The Python Support Forum on digi.com
  www.digi.com/support/forum/categories/python

set rciserial

Purpose

Turns on/off RCI serial mode on the first serial port, and displays current RCI serial-mode settings. The RCI serial mode is a mode that allows a configuration file to be loaded over a serial port when you **assert** or **raise** the **DSR** input signal.

There is no **revert** command variant for this command.

Required permissions

For Digi products with two or more users, you must set permissions to one of the following to use this command:

- For a user to display the RCI serial settings for the line on which they are logged in: "**set permissions s-rciserial=r-self**.
- For a user to display the RCI serial settings for any line: **set permissions s-rciserial=read**.
- For a user to display and set the RCI serial settings for the line on which they are logged in: **set permissions s-rciserial=rw-self**.
- For a user to display the RCI serial settings for any line, and set serial settings for the line on which the user is logged in: **set permissions s-rciserial=w-self-r**.
- For a user to display and set the RCI serial settings on any line: **set permissions s-rciserial=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Turn on off RCI serial mode

```
set rciserial [state={on|off}]
```

Display current RCI serial-mode settings

```
set rciserial
```

Options

state

Enables (on) or disables (off) RCI serial mode on the port. The default is **off**.

Example

```
set rciserial state=on
```

See also

- backup
- show: The **show rciserial** command shows the current RCI serial settings in a Digi device.

set realport

Purpose

Configures and displays RealPort-related settings.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-network=read** to display settings, and **set permissions s-network=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure RealPort settings

```
set realport [keepalive={on|off}]
  [exclusive={on|off}]
  [authentication={on|off}]
   [sharedsecret=string]
```

Display current RealPort settings

```
set realport
```

Options

keepalive={on|off}

Enables or disables sending of RealPort keepalives. RealPort keepalives are messages inside the RealPort protocol, sent approximately every 10 seconds, to tell whoever is connected that the connection is still alive. RealPort keepalives are different from TCP keepalives, which are done at the TCP layer, and configurable. The default is **on**.

As RealPort keepalives generate additional traffic--several bytes every 10 seconds--this option allows you to turn them off. RealPort keepalives may cause issues in environments that are metered for traffic, or that do not require this type of mechanism. In situations such as cellular/mobile wireless communications, when you are paying by the byte, such additional traffic is undesirable when a TCP keepalive can do the same job, and only when the connection is idle.

If you want to have the RealPort keepalive set to **off**, consider using a TCP keepalive instead. If the link is not closed properly, it could result with your port being “locked up” with a dead TCP session.

exclusive={on|off}

Enables or disables exclusive mode for RealPort connections. Exclusive mode allows the Digi device to close an existing RealPort connection and establish a new one immediately upon a new connection request from the same IP address. This mode is useful when using RealPort over wide area networks, which can be unstable and where you are charged by the byte (such as cellular or satellite), and you do not wish to incur costs for keep-alive traffic. Exclusive mode allows your application to retain continuity when temporary, unexpected interruptions in network connectivity occur.

authentication={on|off}

Enables or disables shared-secret authentication between the RealPort client and server.

sharedsecret=*string*

The shared secret is a 1-to-16-character password that is exchanged (in an encrypted form) between the RealPort client and server. If authentication is: Not enabled in both the client and server, not disabled in both, or the shared secret is not the same in both, communications cannot be established.

Example

```
#> set realport keepalive=on
```

See also

- revert: The **revert realport** command reverts the settings configured by this command.
- set autoconnect
- set dirp: This command configures Device-Initiated RealPort connections, where the Digi device initiates RealPort connections, rather than a driver running on Windows, Unix, or Linux initiating the connection.
- set network: The **set network** keepalive options (**idle**, **probe_count**, **probe_interval**, **garbage_byte**, and **override_dhcp**) should be configured for various services that are configured by "**set service keepalive={on|off}**, or clients such as autoconnect (**set autoconnect keepalive={on|off}**").
- set service
- show: The **show realport** command shows the current RealPort settings in a Digi device.
- For ConnectPort X2 gateways, RealPort can be used to directly access the XBee RF module on a gateway. See Direct Access communication with the XBee RF module on ConnectPort X2 gateways.

set realportusb

Purpose

Configures and displays the Multi Host Group assignments; that is, which USB ports are assigned to which Groups.

This command is supported in AnywhereUSB® products only.

Syntax

Configure RealPort USB settings

```
set realportusb [dga={on|off}]
  [port=[item1[,item2...]]
  [group=0-14]
```

Display current RealPort USB settings

```
set realportusb
```

Options

dga={on|off}

Enables/disables the Dynamic Group Assignment (DGA) feature, which lets an administrator change Group assignments on the fly without requiring a reboot of the AnywhereUSB. This means there will be no disruption to unaffected ports.

port=[item1[,item2...]

The port number or range of port numbers to configure. The example

shows several ways of expressing this option.

group=0-14

The group the ports will be assigned. group=0 sets the port to

unassigned.

Example

```
#> set realportusb port=1-4,8,10-12,14 group = 2
```

See also

show: The show realportusb command shows the current RealPort USB settings in an Anywhere USB device.

set rtstoggle

Purpose

Used to:

- Enable or disable RTS toggle on a given serial port. RTS toggle is used to raise RTS when sending data.
- Display current RTS toggle settings.

There is no **revert** command option for reverting the settings configured by this command.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the RTS toggle settings for the line on which they are logged in: **set permissions s-rtstoggle=r-self**.
- For a user to display the RTS toggle settings for any line: **set permissions s-rtstoggle=read**.
- For a user to display and set the RTS toggle settings for the line on which they are logged in: **set permissions s-rtstoggle=rw-self**.
- For a user to display the RTS toggle settings for any line, and set RCI serial settings for the line on which the user is logged in: **set permissions s-rtstoggle=w-self-r**.
- For a user to display and set the RTS toggle settings on any line: **set permissions s-rtstoggle=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Enable or disable RTS toggle

```
set rtstoggle port=range [state={on|off}]
  [predelay=delay]
  [postdelay=delay]
```

Display current RTS toggle settings

```
set rtstoggle [port=range]
```

Options

port=*range*

Used to specify the serial port. Optional on a single-port device.

state={on|off}

Used to enable or disable the RTS toggle feature.

on

Enables the RTS toggle feature.

off

Disables the RTS toggle feature.

The default is **off**.

predelay=*delay*

Specifies the time in milliseconds to wait after the RTS signal is turned on before sending data. The range is **0** to **5000** milliseconds. The default is **0**.

postdelay=*delay*

Specifies the time in milliseconds to wait after sending data before turning off the RTS signal. The range is **0** to **5000** milliseconds. The default is **0**.

Examples

```
#> set rtstoggle state=on predelay=10
```

See also

- revert: The **revert serial** command reverts the settings configured by this command.
- show: The **show rtstoggle** command shows the current RTS toggle settings in a Digi device.

set scancloak

Purpose

Configures the network port scan cloaking feature. This feature allows you to configure this Digi device to ignore (discard) received packets for services that are hidden or not enabled and network ports that are not open.

Malicious software on the Internet may scan IP addresses, protocols and ports to try to gain access to hosts. The network port scan cloaking feature can be used to prevent responses from being sent to the originator for ping and for TCP and UDP ports that do not have an associated service. The default operation is that, when a TCP connection request is received for a port that is not open/bound, the Digi device will send a TCP reset reply to inform the originator that the service is not available. Similarly, the default operation when a UDP datagram is received for a port that is not open/bound, the Digi device will send an ICMP port unreachable packet to inform the originator that the service is not available. For the DNS Proxy feature, specific network interfaces can be configured to ignore (discard) requests that are received from that interface, without otherwise acting on them.

These actions, which are common behaviors in accordance with established protocol standards, effectively inform the originator that it has found a valid IP destination. The originator may continue to probe other ports to gain access to the Digi device. In addition, such reply packets may have a monetary cost for mobile network services (for example, cellular or WiMAX). Enabling the cloaking feature can help manage both the port scanning threat and reduce overall data costs.

Your Digi device can be configured to activate cloaking on a global basis, as well as for individual network interfaces that are available on your device. Activating cloaking on a global basis is configured by setting the **group** option to **global**, which is also the default setting. Enabling cloaking for individual protocols and interfaces is done by specifying the interface name followed by interface-specific options. By enabling cloaking for individual protocols and interfaces, you prevent reply packets from being sent to the originator under the conditions described above.

Note If you enable cloaking on a global basis for a particular protocol, that selection overrides the selections for the interface-specific settings. For example, enabling cloaking for ping in the global group, overrides a disabled selection for the **eth0** (Ethernet) interface.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-scan-cloak=read** to display settings, and **set permissions s-scan-cloak=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

```
set scancloak [state={off|on]
  [group={global|network interface name}
  [group specific options]
  which are:
  [ping={off|on}]
  [tcp={off|on}]
  [udp={off|on}]
  [dns_proxy={off|on}]
```

Note The **dns_proxy** option is not meaningful for the **global** group. Configure the DNS Proxy feature to disable it globally.

Options

state={off|on}

Enables or disables the network port scan cloaking feature on this Digi device.

group={global|*network interface name*}

The group of connection requests to which the command applies, such all connection requests or only Ethernet or mobile connection requests. The valid group names vary according to the network interfaces that are available on your product. For example, for some products, the available interfaces are **{global|eth0|mobile0}**.

To display all available network interfaces on your Digi device that can be configured via the **group** option, enter the **display scancloak** command. Alternatively, the help text displayed for **help set scancloak** will identify the permissible values for the **group** option, according to what is supported in the Digi device. Examples later in this command description demonstrate use of both of these commands to display network interfaces.

If **group** is not specified, the default is **global**, which means that network port scan cloaking is enabled on a global basis.

[*group specific options*]

ping={off|on}

Enables/disables cloaking for ping requests. Replies are not sent for received ping requests.

tcp={off|on}

Enables/disables cloaking for TCP connection requests for which no service is available.

udp={off|on}

Enables/disables cloaking for UDP packets for which no service is available.

dns_proxy={off|on]

Enables/disables cloaking for DNS Proxy requests for a specific network interface.

Note There is no global cloaking selection for DNS Proxy. To cloak the DNS Proxy feature altogether, simply disable it.

Examples

Enable scan cloaking for ping requests on all network interfaces

```
#> set scancloak state=on ping=on
```

Enable scan cloaking on a particular network interface

1. Display all available network interfaces. Use either **display scancloak** or **help set scancloak**.

```
#> display scancloak

Network Port Scan Cloak Status:

  Cloak state: on

  Values configured in the network stack:

                  Ping        TCP         UDP         DNS Proxy
  global          off         off         off         N/A
  eth0            off         off         off         off
  mobile0         off         on          on          on

Network Port Scan Cloak Statistics:

  Packets received but discarded due to cloaking:

                 Ping        TCP         UDP         DNS Proxy
  global         0           175         5           0
  eth0           0           0           0           0
  mobile0        0           175         5           0

#> help set scancloak

syntax: set scancloak [options...]

options:

    state=[off | on]            {scan cloak feature state}
    group=(group_name)
            where (group_name) is one of:
              global, eth0, mobile0
        If group is not specified, the default is "global".
        Note: The valid group name list varies according to the
        network interfaces that are available on your product.

  The following are group-specific options:

    ping=[off | on]               {Ping cloak state}
    tcp=[off | on]                {TCP cloak state}
    udp=[off | on]                {UDP cloak state}
    dns_proxy=[off | on]   {DNS Proxy cloak state}
    Note: The dns_proxy option is not meaningful for the "global" group.
          Configure the DNS Proxy feature to disable it globally.
```

2. The output from both commands shows that the network interfaces for which network port scan cloaking can be enabled are: **eth0**, **mobile0**, and **global**. To enable scan cloaking for all TCP connection requests over the Ethernet interface (**eth0**), enter:

```
#> set scancloak state=on group=eth0 tcp=on
```

See also

- display scancloak
- revert: The **revert scancloak** command reverts the settings set by this command.

- show: The **show scancloak** command shows the network port scan cloaking settings in a Digi device.

set serial

Purpose

Sets and displays general serial configuration options, such as baud rate, character size, parity, stop bits, and flow control.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the serial settings for the line on which they are logged in: **set permissions s-serial=r-self**.
- For a user to display the serial settings for any line: **set permissions s-serial=read**.
- For a user to display and set the serial settings for the line on which they are logged in: **set permissions s-serial=rw-self**.
- For a user to display the serial settings for any line, and set serial settings for the line on which the user is logged in: **set permissions s-serial=w-self-r**.
- For a user to display and set the serial settings on any line: **set permissions s-serial=rw**.

See set permissions for details on setting user permissions for commands.

Permissions for **set serial** also apply to the **set switches** command. See set switches.

Syntax

Set general serial options

```
set serial port=range baudrate=baudrate
 databits={5|6|7|8}
  stopbits={1|2}
  parity={none|odd|even|mark|space}
  flowcontrol={none|software|hardware|custom}
  altpin={on|off}
  closewait={forever|0-600}
  customflow=[{rts|cts|dtr|dsr|dcd|ri|ixon|ixoff}[,...]
  sigsonopen={none|rtsdtr}
```

Display current serial options

```
set serial [port=range]
```

Options

port=*range*

Used to specify the serial port. Optional on a single-port device.

baudrate=*baudrate*

The baud rate in bits per second. The default is **9600**.

databits={5|6|7|8}

The number of data bits used on this line. The default is **8**.

stopbits={1|2}

The number of stop bits per character to use on this line. The value used here must match the setting on the device connected to this port. Use 1 or 2 stop bits.

The default is **1** stop bit.

parity={none|even|odd|mark|space}

The parity used for the line.

none

No parity.

even

Even parity.

odd

Odd parity.

mark

Mark parity.

space

Space parity.

The default is **none**.

flowcontrol={none|software|hardware|custom}

Specifies which kind of flow control is used on the line.

none

No flow control.

software

Software flow control (**Xon**/**Xoff**).

hardware

Hardware flow control (**RTS**/**CTS**).

custom

Custom flow control, as specified by the **customflow** option.

The default is **software**.

altpin={on|off}

The altpin option swaps the pinout position of the **DSR** and **DCD** input signals for the port. Turning altpin on enables access to the **DCD** signal using an eight-wire cable. This option is useful for connecting to modems and for some hardware configurations where carrier detection is important.

on

The altpin option is used.

off

The altpin option is not used.

The default is **off**.

closewait={forever|0-600}

How long a close-port operation waits before it flushes and closes the port. The default is **forever**.

customflow=[{rts|cts|dtr|dsr|dcd|ri|ixon|ixoff}[,...]

The custom flow control used on the line. This option is supported on some but not all Digi devices. This option allows for specifying multiple signals for flow control in non-standard ways, with combinations for the same direction of data transfer; see the example below.

sigsonopen={none|rtsdtr}

Changes the default behavior for various applications so that they will not automatically raise and lower signals when the serial port is opened or a connection is established.

Example

Set baud rate and flow control

```
#> set serial baudrate=9600 flowcontrol=hardware
```

Set custom flow control

This command sets receive flow control to be **RTS** and both **CTS** and **DSR** to be used for transmit flow control.

```
#> set serial customflow=rts,cts,dsr
```

See also

- display serial
- info serial
- revert: The **revert serial** command reverts the settings configured by this command.
- set profile
- set switches
- show: The **show putty** command shows the current serial configuration settings in a Digi device.

set service

Purpose

Used to:

- Enable and disable network services.
- Change the network port on which a given service listens.
- Display the entire service table, or an entry in the service table.

Caution on enabling and disabling services

CAUTION! Exercise caution in enabling and disabling network services, particularly disabling them. Changing certain settings can render your Digi device inaccessible. For example, if you disable Advanced Digi Discovery Protocol (ADDP), the device will not be discovered on a network, even if it is actually connected. If you disable HTTP and HTTPS, the Web interface can be disabled. Disabling basic services such as Telnet, and Rlogin, can make the command-line interface inaccessible.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-service=read** to display settings, and **set permissions s-services=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Enable/disable network services or change network port for service

```
set service [range=range]
  [state={on|off}]
  [ipport=network_port]
  [keepalive={on|off}]
  [nodelay={on|off}]
  [delayed_ack=0-1000]
  [reduced_buffer={on|off}]
```

Display service table or entries in the table

```
set service [range=range]
```

Options

range=*range*

Used to specify the index of the network service to which the rest of the command's options apply. This range varies among Digi devices. Enter **set service** to display the index numbers for the network services on your Digi device. For more information on using this option, see Index numbers and changing default port numbers.

state={on|off}

Used to enable or disable a given network service.

ipport=*network port*

Used to change the network port on which a given network service listens. See Supported network services and their default network port numbers for more information on the network services available.

keepalive={on|off}

Indicates whether or not TCP keepalives will be sent for specified range of network services. If set to on, keepalives will be sent, if it is off, keepalives will not be sent.

Configurable TCP keepalive parameters, for example, how many keepalives to send and when to send them, are configured globally via the **set network** command (see set network).

nodelay={on|off}

Used to allow unacknowledged or smaller-than-maximum-segment-sized data to be sent for the specified range of network services.

The **nodelay** option disables Nagle's algorithm, which is on by default, for some TCP services. The purpose of Nagle's algorithm is to reduce the number of small packets sent. The algorithm establishes not sending outgoing data when there is either unacknowledged sent data, or there is less-than-maximum segment size (typically around 1500 bytes for Ethernet) worth of data to be sent. While this algorithm allows for efficient data transmission, there are times when it is desirable to disable it.

delayed_ack=0-1000

The time, in milliseconds, to delay sending ACK packets in response to received data for the specified range of network services. The default is **200** milliseconds.

Setting this option to **0** (zero) sends an ACK packet back acknowledge the received data immediately. Setting it to any other value means that the ACK packet will be sent after the specified time. If the network services generate new data during that time, the ACK packet will be sent along with the data packet.

You can use this setting to avoid congestion and reduce network traffic, However, do not change this option from its default setting unless you have a solid understanding of network services and data transmission, or have been instructed to make the change.

reduced_buffer={on|off}

The reduced buffer feature limits the amount of data that can be passed through the system to as close to a single byte at a time as is possible for the socket services. This causes throughput to drop considerably. The intended use is for extremely low baud rate applications. Because Digi devices normally buffer a great deal of data, it is possible for a remote client to timeout waiting for the Digi device to complete the transmission at the end of the "session." Artificially limiting the amount of internal buffering by setting **reduced_buffer** to **on** dramatically reduces the amount of time between a remote client requesting that the Digi device close a connection and the Digi device's ability to acknowledge that it is to close the connection.

Supported network services and their default network port numbers

The following table shows the network services controlled by the **set services** command, the services provided, and the default network port number for each service.

In Digi devices that have multiple serial ports, the network port number defaults for various services are set based on the following formula:

base network port number + serial port number

For example, the Telnet Passthrough service is set to network port **2001** for serial port **1**, **2002** for serial port **2**, **2003** for serial port **3**, and so on.

If you change a network port for a particular service, that is the only network port number that changes. That change does not carry over to the other network ports. For example, if you change the network port number Telnet Passthrough from **2001** to **3001**, that does not mean that the other network ports will change to **3002**, **3003**, and so on.

There are two types of network services available:

- Basic services, which are accessed by connecting to a particular well-known network port.
- Passthrough services, in which a particular serial port is set up for a particular type of service. To use the service, users must both use the correct protocol and specify the correct network port. For example, assuming default service ports and using a Linux host, here is how a user would access the SSH and Telnet passthrough services:

```
#> ssh -l fred digi16 -p 2501
```

```
#> telnet digi16 2101
```

Service	Services Provided	Default Network Port Number
ADDP	Advanced Digi Discovery Protocol, also known as Device Discovery. Provides discovery of Digi devices on a network.	2362
Encrypted (Secure) RealPort	Secure Ethernet connections between COM or TTY ports and device servers or terminal servers.	1027
HTTP	Hypertext Transfer Protocol, also known as Web Server. Provides access to web pages for configuration that can be secured by requiring a user login.	80
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer), also known as Secure Web Server. Provides access to web pages for configuration that can be secured by requiring a user login, with encryption for greater security.	443
LPD	Line Printer Daemon. Provides network printing over a serial port.	515
Modem Emulation Pool (pmodem)	Allows the Digi device to emulate a modem. Modem emulation sends and receives modem responses to the serial device over the Ethernet instead of Public Switched Telephone Network (PSTN). Telnet processing can be enabled or disabled on the incoming and outgoing modem-emulation connections. The pmodem service is for connecting to whatever serial port will answer.	50000

Service	Services Provided	Default Network Port Number
Modem Emulation Passthrough	Allows the Digi device to emulate a modem. This service is for dialing in to a particular serial port that has been set up for modem emulation.	50001
RealPort	A virtual connection to serial devices, no matter where they reside on the network.	771
Rlogin	Remote login service. Allows users to log in to the Digi device and access the command-line interface via Rlogin.	513
Rsh	Remote shell service. Allows users to log in to the Digi device and access the command-line interface via Rsh.	514
SSH	Secure Shell service. Allows users secure access to log in to the Digi device and access the command-line interface.	22
SSH Passthrough	Accessing a specific serial port set up for SSH.	2501
Secure Socket Service	Authentication and encryption for Digi devices.	2601
Serial/UDP Server (UDP Passthrough)	Allows raw data to be passed between the serial port and User Datagram Protocol (UDP) datagrams on the network.	2101
SNMP	Managing and monitoring the Digi device through Simple Network Management Protocol. If you want to run SNMP, but in a more secure manner, note that SNMP allows for “sets” to be disabled.This securing is done in SNMP itself, not through this command.	161
TCP Echo	Transmission Control Protocol echo. Used for testing the ability to send and receive over a TCP connection, similar to a ping.	7
Telnet	Allows users an interactive Telnet session to the Digi device’s command-line interface.	23
Telnet Passthrough	Allows a Telnet connection directly to the serial port, often referred to as reverse Telnet.	2001
TCP Passthrough	Transmission Control Protocol passthrough. Allows a raw socket connection directly to the serial port, often referred to as reverse sockets.	2101

Service	Services Provided	Default Network Port Number
UDP Echo	Used for testing the ability to send and receive over a User Datagram Protocol (UDP) connection, similar to a ping.	7
VNC Client Listen Daemon	Remote access to a computer on the network or internet using the VNC (Virtual Network Computing) protocol. VNC server software must be installed on the remote computer.	5500
VNC Server	Allows users to remotely view what is currently displayed on the screen using a standard VNC client (viewer).	5900

Index numbers and changing default port numbers

An index number is assigned to each of these services. The index numbers assigned can vary over time. If you want to change the network port number for a service, enter a **set service** or **show service** command to display the current index number assigned to all services. Locate the service for which you want to change the network port number, and note the index number for the service. Enter a **set service** command, specify that index number for the **range** option, and the new network port number for the **ipport** option.

For example, to change the network port number for the Telnet basic service from its default port number of **23** to **100**, enter the following **set service** command:

```
#> set service
```

The command output displays services defined in and their current network port number assignments:

```
#> set service
Service Configuration :
index state ipport keepalive nodelay dlyd-ack                        service
    1   off      7       off     off      200                TCP Echo Service
    2   off      7        na      na      200                UDP Echo Service
    3    on     22       off     off      200                     SSH Service
    4    on     23       off     off      200                  Telnet Service
   16    on     80        na      na      200                    HTTP Service
   17    on    161        na      na      200                    SNMP Service
    5    on    443        na      na      200                   HTTPS Service
   19   off    513       off     off      200                  Rlogin Service
   20   off    514       off     off      200                     Rsh Service
   13   off    515       off     off      200              Line Printer Daemon
   12    on    771       off      na      200                 RealPort Service
    6    on   1027       off      na      200       Encrypted RealPort Service
    7    on   2001       off     off      200           Telnet Server (Port 1)
    8    on   2101       off     off      200             TCP Server (Port 1)
    9    on   2101        na      na      200       Serial/UDP Server (Port 1)
   18    on   2362        na      na      200                    ADDP Service
   10    on   2501       off     off      200             SSH Server (Port 1)
   11    on   2601       off     off      200  Secure Socket Service (Port 1)
   21   off   4401        on     off      200              Socket Tunnel Server
   14    on  50000        na      na      200            Modem Emulation (Pool)
   15    on  50001       off     off      200          Modem Emulation (Port 1)
```

Note that the index number assigned to the Telnet basic service is **4**. Next, specify **4** for the index number for the **range** option, and the new network port number for the **ipport** option:

```
#> set service range=4 ipport=100
```

Examples

Disable service

```
#> set service range=1 state=off
```

Change the network port (ipport) of a service

```
#> set service range=1 ipport=500
```

Displaying the service table

In this example, the **set service** command displays the entire service table.

```
#> set service
```

Displaying an entry in the service table

In this example, the **set service** command displays a range of entries in the service table.

```
#> set service range=2-4
```

Allow outgoing data that is unacknowledged or less than maximum segment size

```
#> set service ra=5 nodelay=on
```

See also

- revert: The **revert service** command reverts the settings configured by this command.
- set network
- set passthrough for information on network services and applications that are enabled and disabled by default when a Digi device is configured for IP passthrough.
- show: The **show service** command shows the current network service settings in a Digi device.
- For descriptions of Device Cloud and related settings, see set xbee, set mgmtglobal, and set mgmtnetwork.
- For more information on SureLink Link Integrity Monitoring tests, see set surelink.
- For ConnectPort X2 gateways, the RealPort network service can be used to directly access the XBee RF module on the gateway. Doing so requires enabling the RealPort service using this command. See Direct Access communication with the XBee RF module on ConnectPort X2 gateways.

set sharing

Purpose

Configures or displays the port sharing feature. A Digi device enabled for port sharing allows more than one client to open a serial port through RealPort, reverse Telnet, reverse SSH, or connect.

All clients that share a port will read the same data from the serial port; the data is duplicated and sent to each client. All clients that share a port will have the data they write merged and sent out the serial port. The serial port parameters, such as baud rate and flow control, can either be shared by all clients or be controlled exclusively from the Digi device alone.

If there is only one client, RealPort, reverse Telnet, reverse SSH, and connect will work properly.

Port sharing is available on select products only

Port sharing is supported on select Digi products only. To determine whether this command and the port sharing feature are supported on a product, enter **help set ?**. If **sharing** is displayed in the returned list, this command is supported.

About flow control on shared ports

All open shared ports share the same underlying input data buffers, so they must remain roughly in sync in the input data stream. For example, if one client stops reading data, the other clients sharing that same physical port can only read one buffer full of data ahead before they must wait for the first client to catch up.

To overcome this limitation that all clients must remain roughly in sync when reading data, a user-configurable timeout can be set by the **set sharing timeout** option. If one client is waiting for the other clients to read, it only has to wait until the timeout expires and then it will be allowed to continue reading. The other clients, that is, the clients that are not reading data, will lose data from the time the timeout expires until they begin reading again. This timeout will not be set by default.

Considerations and cautions for port sharing

CAUTION! There are several caveats when using port sharing:

- When clients send data to the ports, their data will be intermixed; that is, there is no synchronization of the data. If two clients send data at the same time, the data from one client might appear in the middle of the other client's data.
- If one client stops reading data, the input will be flow-controlled for all clients. Clients can only be able to read data at the rate of the slowest client. (There is a timeout to override this, as described above.)
- Incoming opens, persistent opens, and immediate opens may not behave as expected when multiple clients are opening the port at the same time.
- The modem control lines are not dropped until all clients have closed the port.

- When multiple clients share control of the serial port options, such as baud rate, data size, parity, or flow control, the last options set take effect. The serial port options could be changed unexpectedly by another client. This could leave the RealPort driver confused about the correct serial port settings. Different RealPort drivers might react differently to these unexpected changes in serial port settings.

Configure Serial Port Settings

- Certain applications disable flow control temporarily to drain the serial port when a connection is closed, if hardware flow control is enabled and other applications are sending/receiving data during this time serial overflow errors may occur.
- When multiple clients share control of the serial port options, and a new client opens a port, that new client might momentarily set the options to default values before the application can set the options correctly. This might momentarily disrupt communication with the other clients. Depending on the operating system used by the client, it is possible to set the default serial port options to match the options required by the application, and there will be no disruption.
- When multiple clients share control of the serial-port options, some serial-port options, such as case conversion, carriage return, or new line mapping, might be handled on the client system. Therefore, these options would apply to the client that set these options only.
- When the Digi device exclusively controls the serial port settings, any attempt to change the serial port settings from a client are silently ignored. The client believes the settings have been changed, when in fact they have not. The only way to change the serial port settings is through the command line on the Digi device or through the web UI.
- With reverse Telnet, reverse SSH, and connect, it is possible for a single client machine to open a single shared port multiple times by using multiple telnet or ssh sessions. However, with RealPort, it is not possible for a single client machine to open the same RealPort multiple times and use port sharing. Windows simply prevents one machine from opening a RealPort more than once. Unix does allow a single machine to open a RealPort more than once, but the sharing is happening on that Unix machine, not on the Digi device. Unix sharing does merge data written to the port and shares control of the port. However, it does not duplicate the incoming data to all programs that have opened the same RealPort. Instead, the incoming data is arbitrarily divided among the programs. It is possible for one machine to use port sharing with RealPort, but only by configuring the RealPort driver multiple times for the same Digi device.
- Windows RealPort explicitly forces **DTR** and **RTS** to drop when it closes.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-sharing=read** to display settings, and **set permissions s-sharing=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure port sharing

```
set sharing
  [port=range]
  [clients=maximum clients]
  [control={shared|exclusive}]
  [timeout=timeout]
  [wrpolicy={all|first}]
```

Display current port-sharing settings

```
set sharing
```

Or:

```
show sharing
```

Options

port=*range*

Used to specify the serial port.

Note This is optional on a single-port device.

clients=*maximum clients*

The maximum number of clients that are allowed to share the port. Setting this value to 1 means that port sharing is off; that is, only one client can open the port. Setting this value to **2** means that port sharing is on; that is, the port is enabled to be shared. The maximum value allowed is **32**. There is a slight performance penalty if port sharing is on, even if only one client is using ports. Therefore, this value should be set to **1**, unless port sharing is needed. The default is **1**.

control={shared|exclusive}

Specifies whether control should be shared by all clients, or controlled exclusively by the Digi device.

shared

All clients share control of the serial-port parameters, such as baud rate, data bits, parity, or flow control. Any changes made to these parameters by one client affects all clients.

exclusive

The serial port parameters can only be set from the command line or web UI of the device itself. Any attempt by the clients to change serial-port parameters through RealPort will be silently ignored.

The default is **shared**.

timeout=*timeout*

The flow-control timeout, specified in 1/10ths of a second. This parameter specifies how long a fast client waits for a slower client that has flow-controlled the port (see About flow control on shared ports for more information). After this timeout expires, the faster client is allowed to read ahead in

the data stream, and the slower client begins to lose data. A value of **0** means there is no timeout; the faster client will wait forever if necessary for the slower client and never timeout. A value of **1** means the faster client waits only 1/10 of a second for a slower client; which means essentially no waiting. The maximum value is **6000**. The default is **0**.

wrpolicy={all|first}

Specifies who can transmit on a serial port.

all

Everyone can transmit on the serial port.

first

Only the first module to open the port is allowed to write to the device.

Examples

Display and change port-sharing settings

This example shows using **show sharing** and **set sharing** to display port-sharing settings, configure settings, and display changed settings.

```
#> show sharing

 Port Sharing Configuration :

 port#  clients(max)  clients(cur)     control     timeout    wrpolicy
   1             4             0        shared         0         all
```

Port-sharing parameters are displayed in four columns. **current clients** shows how many clients are currently sharing the port. The **max clients**, **control**, and **timeout** columns show the values set by the **clients**, **control**, and **timeout** options.

Next, a **set sharing** command is issued to change port-sharing parameters:

```
#> set sharing clients=4 control=exclusive timeout=600
#> show sharing
Port Sharing Configuration :

 port#  clients(max)  clients(cur)     control     timeout    wrpolicy
   1             4             0        shared         0         all
#> set sharing clients=1
Warning: Some changes will not take effect until the ports are closed.
#> set sharing

       current       max
tty     clients     clients     control     timeout

  1           2           1      shared         100
  2           0           4   exclusive         600
```

At this point, the two clients disconnect from port **1** and a new client connects to port **2**.

```
#> show sharing

 Port Sharing Configuration :
```

```
port#   clients(max)   clients(cur)      control      timeout     wrpolicy
  1              4              0       shared          0          all
```

The warning message indicates that until the two clients disconnect from port **1**, the **clients** value cannot be reduced to **1**.

See also

- revert: The **revert sharing** option reverts the settings configured by this command.
- show: The **show sharing** command shows the current port-sharing settings in a Digi device.

set slideshow

Purpose

Enables/disables the slideshow feature for displaying images from a USB storage device, and sets the delay before showing each image. When enabled, additional options are available in an on-screen application.

Syntax

```
set slideshow [state={on|off}] [delay=seconds]
```

Options

state={on|off}

Enables/disables the slideshow feature.

on

Enables the slideshow feature.

off

Disables the slideshow feature.

delay=seconds

Number of seconds to delay between showing each image.

Example

Display the current state of the slideshow feature

```
#> set slideshow
```

Enable the slideshow feature and set the delay interval to 10 seconds

```
#> set slideshow state=on delay=10
```

See also

- revert: The **revert slideshow** command reverts the settings configured by this command.
- show: The **show sharing** command shows the current slideshow settings in a Digi device.

set smscell

Purpose

Configures the cellular Short Message Service (SMS) capabilities of the mobile module of the Digi device.

There are several groups of SMS settings, configured by several versions of the **set smscell** command:

- Global settings that configure operation of SMS for the Digi device.
- Settings controlling use of the Sender Control List (SCL) that permits users to select the addresses (or phone numbers) from which SMS messages will be accepted.
- Settings for processing Python commands sent via SMS
- Settings controlling execution of built-in SMS commands. Several built-in commands are supported for execution via SMS messages sent to your Digi device.

Important Notes:

- To determine whether the cellular modem in a Digi device supports SMS, Telnet to the command line and enter the **show smscell** command. If an error message is returned (**error: show option not found**), SMS is not supported for that Digi device.
- SMS is a feature that may be available as part of your mobile service agreement. However, sending and receiving text messages may have additional costs. Before using the SMS capabilities of your Digi device, verify with your mobile service provider that your agreement includes SMS as part of your service plan. Understand the costs of SMS before you enable the SMS features on this Digi device.
- See Supported character set.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-sms-cellular=read** to display settings, and **set permissions s-sms-cellular=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Set global SMS settings

```
set smscell global [state={on|off}]
  [ackrcvdcmds={on|off}]
  [nakpswdfail={on|off}]
  [exteventlog={on|off}]
  [password=password]
  [defreceiver={logonly|python}]
  [cmdidchar=char]
```

Set Sender Control List (SCL) SMS settings

```
set smscell scl [state={on|off}]
  [nakrejectedcmds={on|off}
  SCL entry-specific options; entry is selected with index option:
 index=1-16 (required)
  [enabled={on|off}]
  [address=string]
  [match={exact|right|left|partial}]
```

Set Python-related SMS settings

```
set smscell python state=[on|off}]
  [password=password]
  [readqueuemsgmax=10-100] default=100
  [readholdsecmax=0-86400]
```

Set SMS built-in command settings

```
set smscell command [name={cli|help|cwm|dcloud|idigi|ping}]
  [state={on|off}]
  [password=password]
```

Show all SMS settings

```
set smscell all
```

Options

SMS Global options—"set smscell global"

state={on|off}

Enables or disables cellular SMS. When this option is enabled, the remaining SMS options may be configured. This option is disabled (**off**) by default.

ackrcvdcmds={on|off}

Enables or disables sending an acknowledgment (**ACK**) reply via SMS to the originator of the command message, indicating that the command has been accepted and will be processed. This option is disabled (**off**) by default.

nakpswdfail={on|off}

Enables or disables, when a command is received by SMS, sending a a negative acknowledgment (**NAK**) reply via SMS to the originator of the command message that the command has been rejected because of a password validation failure, such as a missing or incorrect password. This option is disabled (**off**) by default.

exteventlog={on|off}

Enables or disables recording of extended detail (verbose) SMS activity information in the system event log for the Digi device. The SMS feature normally records limited, relevant activities to the system event log. These log entries identify SMS initialization, reconfiguration, and message send/receive activities. This option is disabled (**off**) by default.

For troubleshooting purposes, the message send and receive activity logging can be recorded in greater detail by enabling this option. However, this can result in filling the event log with more SMS

activity records than are useful for normal operation, and it is recommended that this option should be enabled only if greater detail is required for some interval of time.

on

Extended detail (verbose) SMS activity information is recorded in the system event log.

off

Limited relevant SMS activity is logged in the system event log.

password=*password*

The password to submit/accept commands via SMS. When a command message is received via SMS, and a global password is specified in these settings, that password must be provided by the originator of the command message or the message will be rejected by the Digi device. If you configure a command-specific password, that command-specific password must be provided instead of this global command password. Specifically, a command-specific password overrides the global password, and the global password is not considered if a command-specific password is configured in the settings. This option is **disabled** (no global password required) by default.

To remove the password, enter the command

```
set smscell global password=""
```

That is, two consecutive double quote characters designate an empty string, which clears the password.

defreceiver={logonly|python}

The default receiver of SMS messages. When a message is received via SMS, the default message receiver is used to determine which SMS user receives the message and process it. This handling pertains to messages that are not enabled commands for which command processing is performed. The choices for this option are:

logonly

The received message is logged but otherwise not processed (default option).

python

The received message is passed to the standard Python receiver. Further processing of the message text is the responsibility of the Python program that is implemented to receive SMS messages. Note that these messages are logged when they are placed on the Python read queue.

cmdidchar=*char*

The command identifier character. When a message is received via SMS, the first character of that message is examined to determine whether the received message text is actually a command to be performed by the Digi device. The command identifier character selects the character that is used to identify a command. This character can be customized for security or convenience to the user.

The command identifier character can be one of the following characters only:

 # ! % * + /

The default command identifier character is **#**.

Since it is possible that some characters may not pass successfully (or without alteration) through a SMS gateway, the default command identifier character, **#**, can be configured to be one of the alternate characters in the list provided in the web interface. Only one command identifier character can be active at any time.

SMS SCL options—“set smscell scl”

The Sender Control List (SCL) permits user to select the addresses (or phone numbers) from which SMS messages will be accepted. This is in effect a “caller ID” capability in which message senders are screened by the Digi device and either processed or discarded according to the configured SCL rules.

state={on|off}

Enables or disables the Sender Control List capabilities on this Digi device. When this option is enabled, the remaining SCL-specific SMS options can be configured. This option is disabled (**off**) by default.

nakrejectedcmds={on|off}

Enables or disables sending a negative acknowledgment (**NAK**) message reply via SMS if received message is rejected by SCL. If enabled, when a message is received via SMS, SCL is enabled, and the sender is not permitted by the SCL rules, send a negative acknowledgment (**NAK**) message via SMS to the originator of the command message, indicating that the message has been rejected due to the configured SCL rules. This option is disabled (**off**) by default.

For each SCL rule, the following options may be configured:

index=1-16

The index number for SCL entry, ranging from 1 through 16. This index number is required to set entry-specific options.

enabled={on|off}

Enables or disables the use of the rule by SMS. Rules may be enabled and disabled without removing them altogether from the SCL. Disabled rules are ignored when examining received messages.

address=*string*

The address (phone number) of the sender for which this rule applies. If the sender's address matches this configured address, the SMS message is accepted for further processing. If the sender's address does not match any of the enabled SCL rule addresses, it is rejected and no further processing is performed. To remove the address, clear the address field on the settings page.

match={exact|right|left|partial}

The type of address-match test that is to be performed for this rule. There are four supported match types. The address string comparisons are case-independent.

exact

The sender's address must match exactly the address configured for this rule.

right

The sender's address must match the address configured for this rule when comparing the rightmost characters to the shorter of the two strings (sender address, rule address). For example, **5551212** matches **13125551212** since the rightmost characters match to the length of the shorter string, **5551212**. This is the default match type.

The shorter address string must match the rightmost portion of the longer address string.

left

The sender's address must match the address configured for this rule when comparing

the leftmost characters to the shorter of the two strings (sender address, rule address). For example, **1312555** matches **13125551212** since the leftmost characters match to the length of the shorter string, **1312555**.

partial

The sender's address must match the address configured for this rule when comparing the consecutive characters to the shorter of the two strings (sender address, rule address). For example, **312555** matches **13125551212** since the shorter string **312555** is a substring of the longer string **13125551212**.

SMS Python options—"set smscell python"

state=[on|off]

Enables SMS features for Python on this Digi device. When this option is enabled, the remaining Python-specific SMS options can be configured. This option is enabled (on) by default.

password=password

The password to submit/accept Python commands via SMS. This password is checked only if using the #python command form. Although this use is not typical, a message may be directed for deliver to Python by sending **#python** as a command to this Digi device. In such a case, this Python password may be configured to validate the acceptance of such a command message before it is accepted and placed on the dedicated Python SMS message read queue for further processing. When Python is configured as the default message receiver, it is not necessary to use the Digi device command message syntax, since all otherwise unhandled messages will be delivered to the Python read queue. However, password validation is not performed for non-command messages. This option is **disabled** (no Python password required) by default.

To remove the password, enter the command

```
set smscell python password=""
```

That is, two consecutive double quote characters designate an empty string, which clears the password.

readqueuemsgmax=10-100

The maximum number of received messages that can be placed on the dedicated Python SMS message read queue awaiting processing by Python. Once this limit is reached, new received messages are logged but discarded until the read queue falls below this configured maximum message count. The range is **10** to **100** messages. The default value is **100** messages.

readholdsecmax=0-86400

Maximum received message hold time; the maximum amount of time in seconds that a received message will be held on the dedicated Python SMS message read queue while waiting for Python SMS message processing to be brought into service. This setting allows messages to be received and queued for Python before the Python program that processes them is ready to receive such messages, thereby eliminating loss of messages that are received before the Python program is ready to handle them. The range The default value for this setting is **600** seconds (**10** minutes).

SMS built-in command options

Several built-in commands are supported for execution via SMS messages sent to your Digi device. Descriptions of built-in command-related settings for the SMS feature and how to execute them from the command lien follow. Detailed descriptions of the SMS command syntax and supported command options are available on the Digi support site, www.digi.com/support/

name={cli|help|cwm|dcloud|ping}

The command for which to settings are being changed. This is required for setting the command state or password.

cli

Request that a CLI command be run on the Digi device. The output from the CLI command is returned to the sender via SMS, with a limit of around **2000** characters for the number of CLI output characters returned in the reply.

help

The Digi device replies to the sender via SMS with a message that specifies the command syntax and a list of the supported, available commands that may be sent to this device. You may obtain further help for a specific command by sending that command as a parameter. For example, to request a help reply for the **#ping** built-in command, issue the command **#help ping**.

cwm
dcloud
idigi

Manage or obtain status for a device connection to a Device Cloud server. The Digi device replies to the sender via SMS with a message that contains the status or result of the requested action.

Note **cwm** and **idigi** are accepted aliases for **dcloud**.

ping

Request that the Digi device reply to the sender via SMS to verify two-way SMS communication between the sender and the Digi device.

For each built-in command, the following options are supported:

state={on|off}

Enables or disables the specified command for use via SMS. All commands are enabled by default.

password=*password*

The password for submitting or accepting the specified command via SMS. The configured password must be specified on the command message for that message to be accepted for further processing. If a command-specific password is configured, that command-specific password must be provided instead of the global command password (if one is configured, see the SMS Global **password=*password*** option above). Specifically, a command-specific password overrides the global password, and the global password is not considered if a command-specific password is configured in the settings. This option is **disabled** (no command password required) by default.

To remove the password, enter the command

```
set smscell command password=""
```

That is, two consecutive double quote characters designate an empty string, which clears the password.

Supported character set

For SMS via GSM service, it is necessary to translate between the GSM 03.38 7-bit alphabet and ASCII, which is the native character set for the Digi device and is the character set used in the CLI and web UI.

The characters of ASCII and GSM 03.38 do not map one-to-one, and in fact some ASCII characters must be represented in GSM 03.38 as multi-character escape sequences (per extensions to the original GSM 03.38 alphabet). In the table below, such characters are shown as **0x1Bhh** under the **GSM 03.38 Code** column. This notation indicates a two-character sequence, where **hh** is a pair of hexadecimal digits.

In the reverse translation (from GSM 03.38 to ASCII), some of the GSM 03.38 characters have no ASCII counterpart. These are replaced with ASCII space characters. One exception is the **INVERTED QUESTION MARK** (**0x60** in GSM 03.38) that is replaced with an **ASCII QUESTION MARK** (**0x3F**) character.

The following table documents the supported characters and the mapping used between these two alphabets.

Note Unknown characters are replaced with space characters during the translation. In the table below, such characters are shown as **0x20 *** under the **GSM 03.38 Code** column.

Notes for the table:

(1)	The GRAVE ACCENT character (0x60) in ASCII has no counterpart in GSM 03.38. A substitution is made using the APOSTROPHE (0x27) in its place.
*	The characters marked with * indicate a substitution since the ASCII characters have no counterpart in GSM 03.38. These characters are replaced with the SPACE (0x20) character. As such, these characters are not supported in the Digi product support of GSM short messages.

Supported character set for SMS

ASCII Code	GSM 03.38 Code	ASCII Character	Description
0x00	0x20 *	NUL	NULL
0x01	0x20 *	SOH	START OF HEADING
0x02	0x20 *	STX	START OF TEXT
0x03	0x20 *	ETX	END OF TEXT
0x04	0x20 *	EOT	END OF TRANSMISSION
0x05	0x20 *	ENQ	ENQUIRY
0x06	0x20 *	ACK	ACKNOWLEDGE
0x07	0x20 *	BEL	BELL
0x08	0x20 *	BS	BACKSPACE
0x09	0x20 *	HT	HORIZONTAL TABULATION
0x0A	0x0A	LF	LINE FEED
0x0B	0x20 *	VT	VERTICAL TABULATION
0x0C	0x1B0A	FF	FORM FEED

ASCII Code	GSM 03.38 Code	ASCII Character	Description
0x0D	0x0D	CR	CARRIAGE RETURN
0x0E	0x20 *	SO	SHIFT OUT
0x0F	0x20 *	SI	SHIFT IN
0x10	0x20 *	DLE	DATA LINK ESCAPE
0x11	0x20 *	XON	DEVICE CONTROL ONE
0x12	0x20 *	DC2	DEVICE CONTROL TWO
0x13	0x20 *	XOFF	DEVICE CONTROL THREE
0x14	0x20 *	DC4	DEVICE CONTROL FOUR
0x15	0x20 *	NAK	NEGATIVE ACKNOWLEDGE
0x16	0x20 *	SYN	SYNCHRONOUS IDLE
0x17	0x20 *	ETB	END OF TRANSMISSION BLOCK
0x18	0x20 *	CAN	CANCEL
0x19	0x20 *	EM	END OF MEDIUM
0x1A	0x20 *	SUB	SUBSTITUTE
0x1B	0x20 *	ESC	ESCAPE
0x1C	0x20 *	FS	FILE SEPARATOR
0x1D	0x20 *	GS	GROUP SEPARATOR
0x1E	0x20 *	RS	RECORD SEPARATOR
0x1F	0x20 *	US	UNIT SEPARATOR
0x20	0x20	SP	SPACE
0x21	0x21	!	EXCLAMATION MARK
0x22	0x22	"	QUOTATION MARK
0x23	0x23	#	NUMBER SIGN
0x24	0x02	$	DOLLAR SIGN
0x25	0x25	%	PERCENT SIGN
0x26	0x26	&	AMPERSAND
0x27	0x27	'	APOSTROPHE
0x28	0x28	(	LEFT PARENTHESIS
0x29	0x29	)	RIGHT PARENTHESIS
0x2A	0x2A	*	ASTERISK

ASCII Code	GSM 03.38 Code	ASCII Character	Description
0x2B	0x2B	+	PLUS SIGN
0x2C	0x2C	,	COMMA
0x2D	0x2D	-	HYPHEN-MINUS
0x2E	0x2E	.	FULL STOP (PERIOD)
0x2F	0x2F	/	SOLIDUS (SLASH)
0x30	0x30	0	DIGIT ZERO
0x31	0x31	1	DIGIT ONE
0x32	0x32	2	DIGIT TWO
0x33	0x33	3	DIGIT THREE
0x34	0x34	4	DIGIT FOUR
0x35	0x35	5	DIGIT FIVE
0x36	0x36	6	DIGIT SIX
0x37	0x37	7	DIGIT SEVEN
0x38	0x38	8	DIGIT EIGHT
0x39	0x39	9	DIGIT NINE
0x3A	0x3A	:	COLON
0x3B	0x3B	;	SEMICOLON
0x3C	0x3C	<	LESS-THAN SIGN
0x3D	0x3D	=	EQUALS SIGN
0x3E	0x3E	>	GREATER-THAN SIGN
0x3F	0x3F	?	QUESTION MARK
0x40	0x00	@	COMMERCIAL AT
0x41	0x41	A	LATIN CAPITAL LETTER A
0x42	0x42	B	LATIN CAPITAL LETTER B
0x43	0x43	C	LATIN CAPITAL LETTER C
0x44	0x44	D	LATIN CAPITAL LETTER D
0x45	0x45	E	LATIN CAPITAL LETTER E
0x46	0x46	F	LATIN CAPITAL LETTER F
0x47	0x47	G	LATIN CAPITAL LETTER G
0x48	0x48	H	LATIN CAPITAL LETTER H

ASCII Code	GSM 03.38 Code	ASCII Character	Description
0x49	0x49	I	LATIN CAPITAL LETTER I
0x4A	0x4A	J	LATIN CAPITAL LETTER J
0x4B	0x4B	K	LATIN CAPITAL LETTER K
0x4C	0x4C	L	LATIN CAPITAL LETTER L
0x4D	0x4D	M	LATIN CAPITAL LETTER M
0x4E	0x4E	N	LATIN CAPITAL LETTER N
0x4F	0x4F	O	LATIN CAPITAL LETTER O
0x50	0x50	P	LATIN CAPITAL LETTER P
0x51	0x51	Q	LATIN CAPITAL LETTER Q
0x52	0x52	R	LATIN CAPITAL LETTER R
0x53	0x53	S	LATIN CAPITAL LETTER S
0x54	0x54	T	LATIN CAPITAL LETTER T
0x55	0x55	U	LATIN CAPITAL LETTER U
0x56	0x56	V	LATIN CAPITAL LETTER V
0x57	0x57	W	LATIN CAPITAL LETTER W
0x58	0x58	X	LATIN CAPITAL LETTER X
0x59	0x59	Y	LATIN CAPITAL LETTER Y
0x5A	0x5A	Z	LATIN CAPITAL LETTER Z
0x5B	0x1B3C	[	LEFT SQUARE BRACKET
0x5C	0x1B2F	\	REVERSE SOLIDUS (BACKSLASH)
0x5D	0x1B3E	]	RIGHT SQUARE BRACKET
0x5E	0x1B14	^	CIRCUMFLEX ACCENT
0x5F	0x11	_	LOW LINE (UNDERSCORE)
0x60	0x27 (1)	`	GRAVE ACCENT
0x61	0x61	a	LATIN SMALL LETTER A
0x62	0x62	b	LATIN SMALL LETTER B
0x63	0x63	c	LATIN SMALL LETTER C
0x64	0x64	d	LATIN SMALL LETTER D
0x65	0x65	e	LATIN SMALL LETTER E
0x66	0x66	f	LATIN SMALL LETTER F

ASCII Code	GSM 03.38 Code	ASCII Character	Description
0x67	0x67	g	LATIN SMALL LETTER G
0x68	0x68	h	LATIN SMALL LETTER H
0x69	0x69	i	LATIN SMALL LETTER I
0x6A	0x6A	j	LATIN SMALL LETTER J
0x6B	0x6B	k	LATIN SMALL LETTER K
0x6C	0x6C	l	LATIN SMALL LETTER L
0x6D	0x6D	m	LATIN SMALL LETTER M
0x6E	0x6E	n	LATIN SMALL LETTER N
0x6F	0x6F	o	LATIN SMALL LETTER O
0x70	0x70	p	LATIN SMALL LETTER P
0x71	0x71	q	LATIN SMALL LETTER Q
0x72	0x72	r	LATIN SMALL LETTER R
0x73	0x73	s	LATIN SMALL LETTER S
0x74	0x74	t	LATIN SMALL LETTER T
0x75	0x75	u	LATIN SMALL LETTER U
0x76	0x76	v	LATIN SMALL LETTER V
0x77	0x77	w	LATIN SMALL LETTER W
0x78	0x78	x	LATIN SMALL LETTER X
0x79	0x79	y	LATIN SMALL LETTER Y
0x7A	0x20	z	LATIN SMALL LETTER Z
0x7B	0x1B28	{	LEFT CURLY BRACKET
0x7C	0x1B40	|	VERTICAL LINE (PIPE)
0x7D	0x1B29	}	RIGHT CURLY BRACKET
0x7E	0x1B3D	~	TILDE
0x7F	0x20 *	DEL	DELETE

See also

- display smscell
- revert: The **revert smscell** command reverts specified SMS settings groups or all SMS settings.
- show smscell: The **show smscell** shows the current cellular SMS settings in a Digi device.
- smscell: The **smscell** command sends a message to a destination via SMS.

set snmp

Purpose

Configures the Simple Network Management Protocol (SNMP) agent, or displays current SNMP settings. Digi devices support SNMP Versions 1 and 2.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-snmp=read** to display settings, and **set permissions s-snmp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Set SNMP settings

```
set snmp [[trapdestip={ipaddress|FQDN}
  [trapsecdest={ipaddress|FQDN}]]
  [publiccommunity=string]
  [privatecommunity=string]
  [setsenabled={on|off}]
  [authfailtrap={on|off}]
  [coldstarttrap={on|off}]
  [linkuptrap={on|off}]
  [logintrap={on|off}]
```

Display SNMP settings

```
set snmp
```

Options

trapdestip={*ipaddress*|FQDN}

Configures the IP address or, if IPV6 is supported on the device, the fully qualified domain name (FQDN) of the system to which the agent should send traps. To enable any of the traps, specify a non-zero value.

The **trapdestip** option is required in order for alarms to be sent in the form of SNMP traps. See send.

trapsecdest= {*ipaddress*|FQDN}

Configures the IP address or, if IPV6 is supported on the device, the fully qualified domain name (FQDN) of an additional system to which the agent should send traps. To enable the secondary trap destination, a non-zero value for **trapsecdest** must be specified. **trapsecdest** is optional and can be specified only if **trapdestip** is specified.

publiccommunity=*string*

The password required to **get** SNMP-managed objects. The default is **public**.

privatecommunity=*string*

The password required to **set** SNMP-managed objects. The default is **private**.

setsenabled={on|off}

Enables or disables **sets** of SNMP-managed objects.

on

Enables **sets** if the specified private community matches the current private community.

off

Disables **sets** even if the specified private community matches the current private community.

The default is **off**.

authfailtrap={on|off}

Enables or disables the sending of authentication failure traps.

on

Enables the sending of authentication failure traps.

off

Disables the sending of authentication failure traps.

The default is **off**.

coldstarttrap={on|off}

Enables or disables the sending of cold start traps.

on

Enables the sending of cold start traps.

off

Disables the sending of cold start traps.

The default is **off**.

linkuptrap={on|off}

Enables or disables the sending of link up traps.

on

Enables the sending of link up traps.

off

Disables the sending of link up traps.

The default is **off**.

logintrap={on|off}

Enables or disables the sending of login traps.

on

Enables the sending of login traps.

off

Disables the sending of login traps.

The default is **off**.

Examples

Enable authentication failure traps

```
#> set snmp trapdestip=10.0.0.1 authfailtrap=on
```

Specify a new private community string

```
#> set snmp privatecommunity=”StLucia72!”
```

See also

- revert: The **revert snmp** command reverts the settings configured by this command.
- To disable and enable SNMP, use the **set service** command. See set service.
- To disable and enable SNMP alarm traps, see set alarm.
- show: The **show snmp** command shows the current SNMP settings in a Digi device.
- The *User Guide* for your Digi device provides more details on SNMP and the available Management Information Blocks (MIBs) for Digi devices.

set socket_tunnel

Purpose

Configures a socket tunnel. A socket tunnel can be used to connect two network devices: one on the Digi device server's local network and the other on the remote network. This is especially useful for providing SSL data protection when the local devices do not support the SSL protocol.

One of the endpoint devices is configured to initiate the socket tunnel. The tunnel is initiated when that device opens a TCP socket to the Digi device server on the configured port number. The Digi device server then opens a separate connection to the specified destination host. Once the tunnel is established, the Digi device server acts as a proxy for the data between the remote network socket and the local network socket, regardless of which end initiated the tunnel.

The socket tunnel feature is most useful for devices with two interfaces. It could also be used as a connection proxy on a single-interface device, such as the Digi Connect ME. One way the socket tunnel feature would be very useful in a single interface device is when the device has the capability to use specified keys, and other devices connected to it do not have that capability. Using the socket tunnel feature, the device with the key capability basically becomes a security gatekeeper for simple devices that cannot use PKI certificates.

Required Permissions

For Digi products with two or more users, permissions must be set to **set permissions s-socket-tunnel=read** to display settings, and **set permissions s-socket-tunnel=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure a socket tunnel

```
set socket_tunnel [state={disabled|enabled}]
  [timeout={0|seconds}] {0 is no timeout}
  [from_hostname={name|ip address}
  [from_port=port number]
  [from_protocol={tcp|ssl}]
  [to_hostname={name|ip address}]
  [to_port=port number]
  [to_protocol={tcp|ssl}]
```

Display current socket tunnel settings

```
set socket_tunnel
```

Options

state={disabled|enabled

Enables or disables the configured socket tunnel.

timeout={0|*seconds*}] {0 is no timeout}

The timeout (specified in seconds) controls how long the tunnel remains connected when there is no tunnel traffic. If the timeout value is zero, then no timeout is in effect and the socket tunnel stays up until some other event causes it to close.

from_hostname={*name*|*ip address*}

The initiating host: the hostname or IP address of the network device that initiates the socket tunnel.

from_port=*port number*

The initiating port: the port number that the Digi device uses to listen for the initial socket tunnel connection.

from_protocol={tcp|ssl}

The initiating protocol: the protocol used between the device that initiates the socket tunnel and the Digi device. Currently, TCP and SSL are the two supported protocols.

to_hostname={*name*|*ip address*}

The destination host: The hostname or IP address of the destination network device.

to_port=*port number*

The destination port: the port number that the Digi device uses to make a connection to the destination device.

to_protocol={tcp|ssl}

The destination protocol: the protocol used between Digi device and the destination device. Currently, TCP and SSL are the two supported protocols. This protocol does not need to be the same for both connections.

See also

- revert: The **revert socket_tunnel** command reverts the settings configured by this command.
- show: The **show socket_tunnel** command shows the current socket tunnel settings in a Digi device.
- The section on socket tunnel settings in your Digi product's *User Guide*.

set surelink

Purpose

For Digi devices with cellular capability, configures Digi SureLink™ settings. Digi SureLink™ provides an "always-on" mobile network connection to ensure that a Digi Cellular Family device is in a state where it can connect to the network. It does this through hardware reset thresholds and periodic tests of the connection.

There are several groups of command options for set surelink:

- Hardware reset thresholds: these settings can be configured to clear any error states that were resident in the device's cellular module, so the device can once again connect to the network, if the connection is lost. SureLink does this by first resetting the cellular module after a default or specified number of consecutive failed connection attempts, and then resetting the Digi device after a default or specified number of failed consecutive connection attempts. Each of these connection-failure settings can be disabled as well.
- Link integrity monitoring tests: Digi SureLink can be configured to run tests, known as link integrity monitoring tests, that are run over the established PPP connection to ensure that end-to-end communication is possible. There are command options that apply to all link testing, and options for the three available link integrity monitoring tests:
  - Ping Test
  - TCP Connection Test
  - DNS Lookup Test

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-ppp=read** to display settings, and **set permissions s-ppp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Set hardware reset thresholds

```
set surelink index=1-2 (required on dual-SIM devices)
  [module_reset_connect_failures={1-255|0=off}]
  [system_reset_connect_failures={1-255|0=off}]
```

Configure link integrity monitoring tests:

Set general link test options

```
set surelink [state={on|off}]
  [test={ping|tcp|dns}]
  [trigger={interval|idle}]
```

```
  [max_consecutive_failures={1-255|0=off}] (probe failures before link reset)
  [interval=10-65535]
```

Set ICMP ping link test options

```
set surelink [pingaddr1={ipv4 address|fqdn}]
  [pingaddr2={ipv4 address|fqdn}]
```

Set DNS lookup link test parameters:

```
set surelink [dnsfqdn1=dns fqdn]
[dnsfqdn2=dns fqdn]
```

Set TCP connection link test parameters:

```
set surelink [ipaddr1={ipv4 address|fqdn}]
[ipaddr2={ipv4 address|fqdn}]
[ipport=1-65535]
```

Display current SureLink settings

```
set surelink
```

Options

Options for Hardware reset thresholds

index=1-2

The index number of the SIM to which the hardware reset threshold settings apply. This index number is required on dual-SIM devices.

module_reset_connect_failures={1-255|0=off}

The number of failed connection attempts that occur before the cellular modem module is reset. This value can be a number between **1** and **255**, or **0**, which turns off the cellular modem module-reset feature. The default is **3**.

system_reset_connect_failures={1-255|0=off}

The number of failed connection attempts that occur before the Digi Cellular Family device is reset. This value can be a number between 1 and 255, or **0**, which turns off the system-reset feature. The default is **0**, or off.

Options for link integrity monitoring tests

General link test options

state={on|off}

Enables or disables link integrity monitoring tests. If set to **on**, the other link integrity monitoring settings may be configured and are used to verify the functional integrity of the mobile connection. The default is **off**.

test={ping|tcp|dns}

The link integrity monitoring test to be run.

Each test can be used to demonstrate that two-way communication is working over the mobile connection. This variety of tests is provided because different mobile networks or firewalls may allow

or block Internet packets for various services. The appropriate test may be selected according to mobile network constraints and user preference.

The tests are performed only while the mobile connection is established. If the mobile connection is disconnected, the link integrity tests are suspended until the connection is established again.

For the tests to provide meaningful results, the remote or target hosts must be accessible over the mobile connection and not through the LAN interface of the device, if it has one. That is, the settings should be configured to guarantee that the mobile connection is actually being tested.

The link integrity monitoring test settings may be modified at any time. The changes are used at the start of the next test interval.

ping

Ping test. This test uses **ping** (ICMP) as a test to verify the integrity of the mobile connection. The test is successful if a valid ping reply is received in response to the ping request sent. The ping test actually sends up to three ping requests, at three second intervals, to test the link. When a valid reply is received, the test completes successfully and immediately. If a reply is received for the first request sent, there is no need to send the other two requests.

Two destination hosts may be configured for this test. If the first host fails to reply to all three ping requests, the same test is attempted to the second host. If neither host replies to any of the ping requests sent, the test fails. The primary and secondary addresses may be either IP addresses or fully qualified domain names.

tcp

TCP connection test. This test creates a new TCP connection as a test to verify the integrity of the mobile connection. The test is successful if a TCP connection is established to a specified remote host and port number. If the remote host actively refuses the connection request, the test is also considered to be successful, since that demonstrates successful two-way communication over the mobile connection. The TCP connection test waits up to 30 seconds for the connection to be established or refused. When the TCP connection is established, the test completes successfully, and the TCP connection is closed immediately.

Two destination hosts may be configured for this test. If the first host fails to establish (or refuse) the TCP connection, the same test is attempted to the second host. If neither host successfully establishes (or refuses) the TCP connection, the test fails. The primary and secondary addresses may be either IP addresses or fully qualified domain names.

dns

DNS lookup test. This test uses a Domain Name Server (DNS) lookup as a test to verify the integrity of the mobile connection. The test is successful if a valid reply is received from a DNS server. Typically, this means the hostname is successfully “resolved” to an IP address by a DNS server. But even a reply such as **not found** or **name does not exist** is acceptable as a successful test result, since that demonstrates successful two-way communication over the mobile connection. When a valid reply is received, the test completes successfully and immediately.

The DNS servers used in this test for the hostname lookup are the primary and secondary DNS servers obtained from the mobile network when the mobile PPP connection is first established. These addresses may be viewed in the web interface on

the
Administration > System Information > Mobile page.

Note that this DNS lookup test is independent of the normal DNS client configuration and lookup cache, which is used for other hostname lookups. This test has been specifically designed to require communication over the mobile connection for each lookup, and to avoid being "short-circuited" by previously cached information. Also, this test does not interfere in any way with the normal DNS client configuration of this device.

Two hostnames may be configured for this test. If the first hostname fails to get a reply, the same test is attempted for the second hostname. If no reply is received for either hostname, the test fails. The primary and secondary DNS names should be fully qualified domain names. Note that the reverse lookup of an IP address is possible, but that is usually unlikely to succeed in returning a name. Still, such a reverse lookup can be used to demonstrate the integrity of the mobile connection.

trigger={interval|idle}

The conditions under which link integrity monitoring tests are performed.

interval

Link integrity monitoring tests are repeated at the interval specified by the **interval** option.

idle

Link integrity monitoring tests are performed only when idle; that is, if no data is received for the period of time specified by the **interval** option.

This value changes the behavior of the test, in that the test interval varies according to the presence of other data received from the mobile connection.

Although using **trigger=idle** may result in less data being exchanged over the mobile connection, it also prevents the link integrity tests from running as often to verify the true bi-directional state of that connection.

max_consecutive_failures={1-255, 0=off} (probe failures before link reset)]

The maximum number of consecutive Link Integrity Monitoring tests. After this number is reached, the mobile connection should be disconnected and reestablished.

This value must be between **1** and **255**. The default is **3**. A value of **0** turns off this feature. When the mobile connection is reestablished, the "consecutive failures" counter is reset to zero.

Note If the mobile connection is disconnected for any reason (including not as a result of a link integrity test failure), the consecutive failures count is reset to zero when the mobile connection is reestablished.

interval=10-65535

Specifies the interval, in seconds, at which the selected Link Integrity Monitoring test is initiated or repeated. A new test will be started every specified number of seconds while the mobile connection is established. This value must be between **10** and **65535**. The default is **240**.

If the configured interval is less time than it takes a test to complete, the next test will not be initiated until the previous (current) test has completed.

ICMP ping link test options

pingaddr1={*ipv4 address*|*fqdn*}
The first host to test.
pingaddr2={*ipv4 address*|*fqdn*}
The second host to test, if the first host fails.
TCP connection link test parameters:
ipaddr1={*ipv4 address*|*fqdn*}
The first host to test.
ipaddr2={*ipv4 address*|*fqdn*}
The second host to test, if the first host fails.
ipport=1-65535
The TCP port number to connect to on the remote host. The default is **80**.
DNS lookup link test parameters
dnsfqdn1=dns fqdn
The first hostname to look up.
dnsfqdn2=dns fqdn
The second hostname to look up, if the first hostname fails.

Example

```
set surelink system_reset_connect_failures=30
```

See also

- The **display pppstats** command displays connection and activity information for PPP links, including SureLink statistics. See SureLink Statistics for descriptions of these statistics.
- revert: The **revert surelink** command reverts the settings configured by this command.
- set mobile
- set mobileppp
- show: The **show surelink** command shows the current Digi SureLink settings in a Digi device.
- *Digi SureLink™ "Always-On" Connection White Paper*, available from the **Documentation** page for Digi Cellular Family products at www.digi.com.

set switches

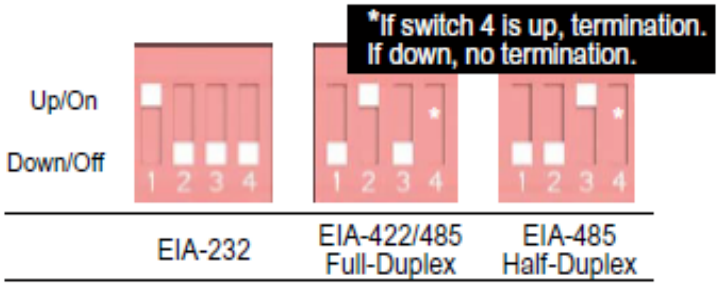
Purpose

For Digi devices with Multiple Electrical Interface switch-setting capability, configures MEI settings on a per-port basis, and displays current MEI settings. MEI settings include the type of electrical interface (EIA-232 or EIA-485), the number of differential wires used for communication, and whether termination and biasing resistors are used.

MEI 232/422/485 switch settings

The figure shows the DIP switches that set the MEI (multiple electrical interface) line protocol on the serial interface. The table below the figure shows the direction to set the switch for each MEI function.

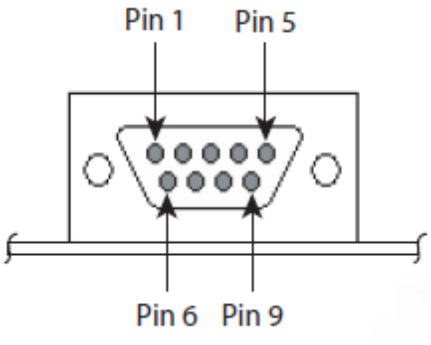
Note This information regarding DIP switches applies to MEI-equipped Digi devices with up to 4 ports. For MEI-equipped Digi devices with 8 or 16 ports, MEI settings can be changed through software only, using this **set switches** command or the web interface **Configuration > Serial Ports > Multiple Electrical Interface (MEI) Serial Settings**.



Function	Switch Setting			
	1	2	3	4
EIA-232	Up/On	Down/Off	Down/Off	Down/Off
EIA-422/485 full-duplex	Down/Off	Up/On	Down/Off	If up, termination. If down, no termination
EIA-485 half-duplex	Down/Off	Down/Off	Up/On	

DB-9 connector and pin orientation

The figure shows the DB-9 pin orientation. The table shows pin signal assignments.



DB9 pinouts

DB-9 pin assignments			
Pin	**EIA-232**	**EIA-422/485 full-duplex**	**EIA-485 half-duplex**
1	DCD	CTS-	Not used
2	RXD	RXD+	RXD+
3	TXD	TXD+	TXD+
4	DTR	RTS-	Not used
5	GND	GND	GND
6	DSR	RXD-	RXD-
7	RTS	RTS+	Not used
8	CTS	CTS+	Not used
9	NA	TXD-	TXD-

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display switch settings for the line on which they are logged in: **set permissions s-serial=r-self**.
- For a user to display switch settings for any line: **set permissions s-serial=read**.
- For a user to display and set the switch settings for the line on which they are logged in: **set permissions s-serial=rw-self**.
- For a user to display the switch settings for any line, and set switch settings for the line on which the user is logged in: **set permissions s-serial=w-self-r**.
- For a user to display and set the switch settings on any line: **set permissions s-serial=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Configure MEI switches

```
set switches [port=range]
  [mode={232|485}]
  [wires={two|four}]
  [termination={on|off}]
```

Display current MEI switch settings

```
set switches
```

Options

range=*range*

The port or range of ports to which this command applies.

mode={232|485}

Selects the electrical interface of the serial port. The selected value determines whether the **wires** and **termination** options are meaningful.

232

The serial port uses electrical interface EIA-232. This interface uses independent wires to transmit and receive data, which allows data to be sent and received between devices simultaneously.

485

The serial port uses electrical interface EIA-485. This mode can also be used for EIA-422 connections. This interface uses either two or four wires to transmit and receive data, depending on the setting of the wires setting. This interface also allows for multiple transmitters and receivers to be easily connected together.

The **wires** and **termination** command options specifically apply to serial ports in EIA-485 mode.

The default is **232**.

See Examples for information on setting and DB-9 connector and pin orientation for DB-9 connector pin orientation and pinouts.

wires={two|four}

Applies when the serial port is running in EIA-485 mode only. Selects the number of differential wires used for communication and implicitly determines the duplex of the connection.

two

The serial port operates in two-wire mode. This mode is a half-duplex connection with *shared* transmit and receive wires.

four

The serial port operates in four-wire mode. This mode is a full-duplex connection with *independent* transmit and receive pairs.

The default is **four**.

termination={on|off}

Applies when the serial port is running in EIA-485 mode only. Determines whether termination and biasing resistors are used across the lines.

on

Termination and biasing resistors are enabled across the lines.

Select termination to **on** if termination or use of biasing resistors is needed across the lines. Enable this setting if the terminal server port is an endpoint node of the EIA-422/485 network and termination or biasing is desired. If using 2-wire mode, termination or biasing is used at one of the end points; usually the Master endpoint.

Note The **CTS** and **RTS** control signals are available as separate differential signals in the EIA-422/EIA-485 4-wire mode. Do not use these differential signals in 2-wire mode. The **CTS** and **RTS** differential signals are not terminated or biased internally. If termination or biasing is needed, it must be done externally.

off

Termination and biasing resistors are disabled across the lines.

The default is **off**.

Examples

Configure standard EIA-232 communication

```
#> set switches port=1 mode=232
```

Configure a half-duplex EIA-485 endpoint

```
#> set switches port=1 mode=485 wires=two termination=on
```

Configure a full-duplex 422 interior node

```
#> set switches port=1 mode=485 wires=four termination=off
```

See also

- revert: The **revert switches** command reverts the settings configured by this command.
- set serial and the **altpin={on|off}** option.
- show: The **show switches** command shows the current MEI switch settings in a Digi device.

set system

Purpose

Configures and displays system-identifying information, such as a description of the device, its location, and a contact person.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-system=read** to display settings, and **set permissions s-system=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Change system-identifying information

```
set system [description=string]
[location=string]
[contact=string]
```

Display system-identifying information

```
set system
```

Options

description=*string*

A description of this device. The maximum length is **64** characters. The default is **“”** (blank description field).

location=*string*

The location of this device. The maximum length is **64** characters. The default is **“”** (blank location field)

contact=*string*

The contact for this device. The maximum length is **64** characters. The default is **“”** (blank contact field).

Examples

Set description, contact, and location

```
#> set system description=”Engineering printer” location=”Room 1347”
contact=”John Doe at x-3749”
```

See also

- info device
- revert: The **revert system** command reverts the settings configured by this command.
- show: The **show system** command shows the current system-identifying settings in a Digi device.

set tcpserial

Purpose

Used to set behaviors of TCP serial connections, or display current TCP serial settings.

This command affects the following TCP serial connections:

- Connections made using the autoconnect feature.
- Incoming network connections made to the following:
  - The TCP server (raw socket, IP port 2101)
  - The Telnet server (telnet socket, IP port 2001)
  - Secure Sockets Layer (ssl socket, IP port 2601)

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the TCP serial settings for the line on which they are logged in: **set permissions s-tcpserial=r-self**.
- For a user to display the TCP serial settings for any line: **set permissions s-tcpserial=read**.
- For a user to display and set the TCP serial settings for the line on which they are logged in: **set permissions s-tcpserial=rw-self**.
- For a user to display the TCP serial settings for any line, and set TCP serial settings for the line on which the user is logged in: **set permissions s-tcpserial=w-self-r**.
- For a user to display and set the TCP serial settings on any line: **set permissions s-tcpserial=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Set behaviors of TCP serial connections

```
set tcpserial port=range
  [hangupdcd={on|off}]
  [hangupdsr={on|off}]
  [idletimeout={0-65000}]
  [sid={on|off}]
  [sidstring=socketid string]
```

```
Buffered forwarding options:
  [buffered={on|off}]
  [sendcount=1-65535 bytes]
  [sendtime={0|1-65535ms}]
  [endpattern=string]
  [strippattern={on|off}
```

Display TCP serial settings

```
set tcpserial [port=range]
```

Options

port=*range*

Used to specify the serial port. Optional on a single-port device.

hangupdcd={on|off}

Indicates whether an established network connection should be terminated when the serial port's DCD signal drops. The default is **off**.

hangupdsr={on|off|}

Indicates whether an established network connection should be terminated when the serial port's DSR signal drops. The default is **off**.

idletimeout={0-65000}

Indicates that established network connection should be terminated if the serial port is idle for the specified amount of time in seconds. A value of **0** (zero) disables this option. The default is **0**.

sid={on|off}

Determines how the socket ID (SID) string in the **sidstring** option is handled.

on

The value for the **sidstring** option is sent to the network destination right before the first data bytes are sent to the network.

off

The value for the **sidstring** option is not sent to the network destination.

The default is **off**.

sidstring=*socketid string*

When the **sid** option is set to on, this string is sent to the network destination right before the first data bytes are sent to the network. The maximum length of this string is **256** characters, including escape sequences for special characters. The maximum parsed length of this string is **256** characters. That is, this string must reduce down to a 256-character string when the escape sequences are processed. For more details on the escape sequences, see Entering Special Characters in String Values.

buffered={on|off}

Turning on this feature on allows controlling how serial data is sent out to the network. The **sendcount**, **sendtime**, **endpattern**, and **strippattern** options are used to control how data is sent out once the **buffered** option is set to **on**. The default is **off**.

sendcount=1 - 65535 bytes

Indicates that data from the serial port should be sent out to the network after buffering the given number of bytes. This option only is valid when the **buffered** option is **on**. The default is **1024** bytes.

sendtime={0|1-65535ms}

Indicates that data from the serial port should be sent out to the network after the given amount of time has past where no new data has arrived from the serial port. This option only is valid when the **buffered** option is **on**. A value of **0** (zero) disables this option. The default is **0**.

endpattern=string

Indicates that data from the serial port should be sent out to the network after the given endpattern string has been found in the data from the serial port. This option only is valid when the **buffered** option is **on**. An empty string disables this option.

The maximum length of this string is **16** characters, including escape sequences for special characters. For more details on the escape sequences, see Entering Special Characters in String Values. The maximum parsed length of this string is **4** characters. That is, this string must reduce down to a 4-character string when the escape sequences are processed.

strippattern={on|off}

This option corresponds with the **endpattern** option. When a valid **endpattern** string is found, this option indicates whether the matching string is stripped or kept in the data stream. The default is **off**.

Examples

```
#> set tcpserial hangupdcd=off idletime=20
```

```
#> set tcpserial port=1 sid=on sidstring=”abc”
```

```
#> set tcpserial port=1 buffered=on sendtime=50 sendcount=512
```

```
#> set tcpserial
```

See also

- revert: The **revert tcpserial** command reverts the settings configured by this command.
- show: The **show tcpserial** command shows the current TCP serial connection settings in a Digi device.

set term

Purpose

Allows for connecting a terminal to a device's serial port and accessing the command line of the device.

In the cases where the default access to the terminal and the command line is **on**, this command is important if you want to use the serial port for purposes other than having a command line. That is, to use the serial port for another purpose, you must change the state of the serial port access from **on** to **off**.

To make sure that changes to terminal settings take effect, reboot the Digi device is recommended after entering this command.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-term=read** to display settings, and **set permissions s-term=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure terminal settings

```
set term [state={on|off}]
```

Display current terminal settings

```
set term
```

Options

state={on|off}

Specifies whether terminal access is enabled for the serial port. For Digi Connect WAN devices, the default is **on**.

Examples

```
#> set term

 Serial Terminal Configuration :

 port# state
     1    on
     2    on
```

See also

- revert: The **revert term** command reverts the settings configured by this command.
- show: The **show snmp** command shows the current terminal device connection settings in a Digi device.

set time

Purpose

Sets the Coordinated Universal Time (UTC) and/or system time and date on a Digi device.

If the **offset** option is set to anything other than **00**, this command assumes that if date and time are being set, they are system time.

Out of the box, all Digi devices maintain time and date as the UNIX epoch (**00:00:00, January 1, 1970**) plus device up-time. On devices with no real-time clock (RTC), date and time revert to the UNIX epoch on each reboot or power-cycle. On devices with a RTC, date and time is the UNIX epoch plus time since initial power-up.

You can set device time manually using any of the usual human interfaces, such as the command line interface or web interface, or it can be set and maintained using a clock source. See set clocksource.

On a Digi device with no real-time clock and no configured clock source, time and date are completely local to the device and have limited usefulness since they are not persistent over reboots/power-cycles.

On a device with a RTC and no configured clock source, time and date are also local to the device, but they are meaningful because they are persistent. The **offset** option could be useful in adjusting for daylight savings time. Setting the date and time to standard time and setting **offset** to **1** whenever daylight savings time is in effect serves that purpose.

On a device with a configured clock source, time and date received from a clock source are expected to be UTC. For users with several devices in different time zones, keeping **offset=0** might be useful for comparing logs or traces from different devices, since all would be using UTC.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-time-source=read** to display current date and time, and to **set permissions s-time-source=rw** to set the date/time and configure related settings. For devices with a real time clock (RTC), **set permissions s-rtc=rw** permissions is required. See set permissions for details on setting user permissions for commands.

Syntax

Configure UTC and/or system time

```
set time [time={hh:mm:ss|hh:mm}]
  [date=mm.dd.yy]
  [offset={hh:mm|hh}]
```

Display current time and date

```
set time
```

Options

time={*hh*:*mm*:*ss*|*hh*:*mm*}

The time. Hours can range from **00** to **23**, minutes can range from **00** to **59**, and seconds can range from **00** to **59**. If omitted, all default to **00**.

date=*mm.dd.yy*

The date. Month can range from **01** to **12**, day can range from **01** to **28**, **29**, **30**, or **31**, depending on the month and leap year, and year can range from **00** to **36**, representing the years **2000** through **2036**.

offset={*hh*:*mm*|*hh*}

The offset from date and time, in hours and minutes. Offset can range from **-12** hours to **14** hours. Very rarely, a time zone can also have an offset in minutes (**15**, **30**, or **45**). If minutes are not specified, the default is **00**. This value can be used to modify the time and date (generally expected to be UTC) to compensate for time zones and daylight savings time. The offset value is added to internal date and time whenever time is provided by the device.

Example

Set the date and time to 2:15 PM, April 3, 2008

```
#> set time time=14:15 date=04.03.08
```

Set the offset for Central Standard Time

```
#> set time offset=-6
```

Set the offset for Central Daylight Time

```
#> set time offset=-5
```

See also

- info time
- revert: The **revert time** command reverts the settings configured by this command.
- set clocksource
- set timemgmt
- show: The **show time** command shows the current time settings in a Digi device.

set timemgmt

Purpose

Configures time source management settings.

Required permissions

For Digi products with two or more users, permissions must be set to **s-time-source=read** to display time source management settings, and to **s-time-source=rw** to display and configure them. See set permissions for details on setting user permissions for commands.

Syntax

Configure time source management settings

```
set timemgmt [adjust_threshold=(seconds, 0 - 300)]
[recovery_state={on|off}]
```

Options

adjust_threshold=*seconds*

The Time Adjustment Threshold. This is a value in seconds that defines a range around the current time value maintained by the device. If a time update is received from a best-ranking (smallest-value) time source and the new time is within that range, the device's time is not changed. However, if the new time falls outside the defined threshold range, the device's time is updated immediately using the new time value.

The Time Adjustment Threshold value can range from **0** to **300** seconds. The default is **60**. For example, if the configured threshold is **60** seconds, the device's time will be updated using a new time value that is **60** seconds or more different than the device's current time value. If the new time value differs from the device's current time by less than **60** seconds, the device's time is not updated using that new time.

The Time Adjustment Threshold is useful in limiting the amount of drift that will be tolerated before the device's time is updated using a new sample. An appropriate value should be selected with consideration for the reliability of the time sample sources.

In the case of NTP/SNTP server sources, the latency, round-trip timing and reliability of the network connection (between the device and the server) also should be considered.

For example, if the communications path between device and server involves a cellular network connection, the performance and behavior characteristics of the cellular network should be taken into account. In a cellular network, intermittent packet delays are possible in either the transmit or receive direction (or both). Frequently these delays are asymmetric, such that the delay is greater in one direction than in the other.

In such conditions, the round trip timing (of the request/reply) skews the time sample adjustment to determine the time value to use for the device. Therefore configuring an aggressively small (short) threshold value may cause the device to adjust its time frequently and unnecessarily, such that the time value “jumps” forward or backward as a consequence of asymmetric packet delays.

recovery_state={on|off}

The state of the Enable Lost Time Source Recovery setting. If multiple external time sources are available and configured, either by the **set clocksource** command or the Time Source Settings in the web interface, normally only the best-ranking (smallest value) source will be used to maintain the device's time. If the best-ranking source stops reporting new time values, it is considered “lost."

on

Enables Lost Time Source Recovery, which allows one or more worse (higher value) ranking time sources to be consulted in an effort to obtain a fresh time value. This prevents the best-ranking configured time source from blocking time updates if that source stops providing acceptable time samples.

The interval of time that must pass for Lost Time Source Recovery to begin varies according to the best-ranking time source that is reporting a value. For a time source of type **sntp** (an SNTP server, set by **set clocksource type=sntp**), the missing sample update interval is three NTP/SNTP intervals configured for that time source, plus one minute. For a time source other than **sntp**, the missing sample update interval is 61 minutes. These interval values cannot be user-configured.

off

Disables Lost Time Source Recovery.

Example

Set the adjustment threshold to 5 seconds and enable lost time source recovery

```
#> set timemgmt adjust_threshold=5 recovery_state=on
```

See also

- info time
- revert: The **revert timemgmt** command reverts the settings configured by this command.
- set clocksource
- set time

set trace

Purpose

Configures a Digi device server for tracing and displays tracing information.

WARNING! Important: The **set trace** command should be used when working with Digi Technical Support. Enabling tracing can have an impact on system performance. Digi provides no guarantee that trace output is the same across firmware revisions.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-trace=read** to display tracing information, and to **set permissions s-trace=rw** to display tracing information and configure trace options. See set permissions for details on setting user permissions for commands.

Syntax

Display latest command options

The syntax and available options for **set trace** vary by product and product release. Enter the following command to view the current list of options:

```
help set trace
```

Configure trace options

```
set trace [state={off|on|dump}]
[mode={historical|concurrent}]
[syslog={on|off}
[loghost=ip address]
[mask=type:severity]
```

Display tracing information

```
set trace
```

Options

state={off|on|dump}

Sets the state of the tracing function.

off

Turns the tracing function off.

on

Turns the tracing function on.

dump

Displays historical trace messages, when **mode** is set to **historical**.

mode={historical|concurrent}

Sets handling of trace messages.

historical

All trace messages stored in the buffer are displayed by issuing the command:

```
#> set trace state=dump
```

concurrent

All trace messages are printed to the administrative terminal.

syslog={on|off}

Enables or disables sending trace messages to the syslog server identified by the **loghost=*ip address*** option.

loghost=*ip address*

The IP address of a host to which trace messages should be sent. This host must be running the **syslog** daemon.

mask=*type*:*severity*

Identifies the type and nature of events that should be traced, and the severity level of the events.

type

The type of events that should be traced. Available ***type*** keywords vary among Digi devices. Enter **set trace ?** to view the list of types supported in the Digi device. Some commonly used trace types for diagnosing connection problems are **modem** and **ppp**. Contact Digi Technical Support for assistance in using the appropriate ***type*** keyword.

***severity*={assert|critical|warning|info|debug}**

The severity level of events traced.

assert

Tracing applies to assert lines only. This severity level is for Digi-internal use only.

critical

Tracing applies to only the most severe events. This is the default severity level. This level produces the least amount of trace data.

warning

Tracing applies to critical events and on less severe events as well. This level produces more trace data than **critical**, but less than **info**.

info

Tracing applies to many events. It produces more trace data than assert, critical, and warning levels.

debug

Used for Digi-internal debugging purposes only.

Examples

```
#> set trace

trace is currently off, using historical mode
syslog is currently off, loghost is 0.0.0.0
system  : ac___     ADDP    : ac___     fwupd   : ac___     config  : ac___
printf  : ac___     inetd   : ac___     simplepw: ac___     webui   : ac___
snmp    : ac___     rciser  : ac___     portsw  : ac___     connect : ac___
sertcp  : ac___     i2c     : ac___     pmodem  : ac___     acl     : ac___
router  : ac___     nat     : ac___     edp     : ac___     dhcps   : ac___
ddns    : ac___     alarm   : ac___     user1   : ac___     user2   : ac___
user3   : ac___     user4   : ac___     user5   : ac___     user6   : ac___
user7   : ac___     user8   : ac___     user9   : ac___     user10  : ac___
ppp     : ac___     chat    : ac___     ssh     : ac___     modem   : ac___
ssi     : ac___     dns     : ac___     vpn     : ac___     treck   : ac___
address : ac___     usb     : ac___     pci     : ac___     tftp    : ac___
surelink: ac___     st      : ac___     ssl     : ac___
```

Output

See www.digi.com/support for descriptions and interpretations of trace output. Digi provides no guarantee that trace output is the same across firmware revisions.

See also

- revert: The **revert trace** option reverts the settings configured by this command.
- display logging
- display techsupport
- The **info** commands. These commands display various device statistics that may aid in troubleshooting your Digi product.
- The Troubleshooting chapter of the *User Guide* for your Digi product.
- The Digi Support web page, to contact Technical Support, search Digi's knowledge base, ask a question on the Support forum, and get diagnostics and utilities.

set udpserial

Purpose

Use this command to set up the UDP serial feature, or display current UDP serial settings.

The UDP serial feature allows you to send data between the serial port and one or more remote network destinations using the UDP protocol. When this feature is enabled for a given serial port, data sent to the serial port will be sent out to the configured destinations. Also any time data is sent to the UDP serial service (IP port) and the serial port is not being used by another service, the data is sent to the serial port **2101**.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the UDP serial settings for the line on which they are logged in: **set permissions s-udpserial=r-self**.
- For a user to display the UDP serial settings for any line: **set permissions s-udpserial=read**.
- For a user to display and set the UDP serial settings for the line on which they are logged in: **set permissions s-udpserial=rw-self**.
- For a user to display the UDP serial settings for any line, and set UDP serial settings for the line on which the user is logged in: **set permissions s-udpserial=w-self-r**.
- For a user to display and set the UDP serial settings on any line: **set permissions s-udpserial=rw**.

See set permissions for details on setting user permissions for commands.

Syntax

Set general UDP serial forwarding characteristics for a serial port

```
set udpserial port=range [state={on|off}]
  [sendcount=bytes]
  [sendtime={0|time}]
  [endpattern=string]
  [strippattern={on|off}]
  [sid={on|off}]
  [sidstring=string}
  [closetime=time]
```

Set UDP destinations for a given serial port

```
set udpserial port=range range=1-64
  [description=string]
  [active={on|off}
  [ipaddress=ip address}]
  [ipport=ip port]
```

Display current UDP serial settings

```
set udpserial [port=range [range=range]]
```

Options

Options for setting general UDP serial forwarding characteristics

port=*range*

Used to specify the serial port. Optional on a single-port device.

state={on|off}

Used to enable or disable sending data from the serial port to remote network destinations. The default is **off**.

sendcount=*bytes*

The number of bytes received from the serial port that causes the data to be sent on to the network destinations. This trigger cannot be disabled. The default is **1024** bytes.

sendtime={0|*time*}

The amount of idle time, in milliseconds, allowed before sending data to the network. If no data is received on the serial port for the time specified by this option, any buffered data is sent on to the network destinations. A value of **0** (zero) disables this trigger.

endpattern=*string*

If this string is set, any pattern match of data received from the serial port causes the data to be sent on to the network destinations. The maximum length of this string is **16** characters, including escape sequences for special characters. For more details on the escape sequences, see Entering Special Characters in String Values. The maximum parsed length of this string is **4** characters. That is, this string must reduce down to a 4-character string when the escape sequences are processed.

strippattern={on|off}

Determines how the data specified by the **endpattern** option is handled.

on

The endpattern that is found is stripped from the stream before any data is to be sent on to the network destinations.

off

The endpattern is not stripped from the stream before data is sent on to network destinations.

The default is **off**.

sid={on|off}

Determines how the socket ID (SID) string in the **sidstring** option is handled; that is, whether the string specified by the **sidstring** option is sent at the beginning of each UDP packet.

on

The value of **sidstring** is sent at the beginning of each UDP packet.

off

The value of **sidstring** is not sent at the beginning of each UDP packet.

The default is **off**.

sidstring=*string*

The string sent at the beginning of each UDP packet if the "sid" option is set to on. The maximum length of this string is **256** characters, including escape sequences for special characters. For more details on the escape sequences, see Entering Special Characters in String Values. The maximum parsed length of this string is **256** characters. That is, this string must reduce down to a 256-character string when the escape sequences are processed.

closetime=*time*

The amount of idle time before closing the serial port, in milliseconds. If no data is sent or received on the serial port for the specified amount of time, the serial port is closed. This allows the serial port to be used by other things such as TCP socket or RealPort. If a value of **0** is set, the **closetime** option recalculates internally to be 1000 milliseconds or twice the send time, whichever is greater. The default is **0** milliseconds.

Options for setting UDP destinations for a given serial port

The following options require a specific range to be specified by the **range** option.

port=*range*

Specifies the serial port. Optional on a single-port device.

range={1-64}

Specifies the UDP destination to be configured.

description=*string*

A string for descriptive purposes only.

active={on|off}

Specifies whether data from the serial port is sent to this destination.

on

Data from the serial port is sent to this destination.

off

This destination is not sent any data.

The default is **off**.

ipaddress=*ip address*

The IP address of the network destination to which data is sent.

ipport=*ip port*

The UDP port of the destination to which data is sent.

Options for displaying current UDP serial settings

port=*range*

Used to specify the serial port. Optional on a single-port device.

range=range

Identifies the range of UDP destinations to be displayed.

Examples

Set general UDP serial forwarding based on bytes received

In this example, the amount of bytes received from the serial port causes the data to be sent on to the network destination:

```
#> set udpserial port=1 state=on sendcount=2
```

Set UDP destinations for a given serial port

In this example, data is sent to the specified destination:

```
#> set udpserial port=1 range=1 ipaddress=10.0.0.1 ipport=2101 active=on
```

Display current UDP serial settings

The following are all valid ways of using **set udpserial** to display current UDP serial settings:

```
#> set udpserial
```

```
#> set udpserial port=1
```

```
#> set udpserial port=1 range=1-12
```

See also

- revert: The **revert udpserial** command reverts the settings configured by this command.
- show: The **show udpserial** command shows the current UDP serial settings in a Digi device.

set user

Purpose

Used to:

- Add users for access to a Digi device. The number of users that can be defined varies by Digi device. To determine the number of users allowed for your Digi device, enter **set user** or **show user**.
- Associate a user with a group. A user can be associated with up to two groups.
- Disassociate a user from a group.
- Remove users.
- Change user configuration attributes.
- Display user configuration attributes.
- Load an SSH public key, and, for single-user model products, unload a public key.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to one of the following:

- For a user to display the user settings for the line on which they are logged in: **set permissions s-user=r-self**.
- For a user to display the user settings for any line: **set permissions s-user=read**.
- For a user to display and set the permissions settings for the line on which they are logged in: **set permissions s-user=rw-self**.
- For a user to display the permissions settings for any line, and **set group** settings for the line on which the user is logged in: **set permissions s-user=w-self-r**.
- For a user to display and set the user settings on any line: **set permissions user=rw**. When permissions are set to **set permissions s-user=rw**, a user cannot set another user's permission level higher than their own level or raise their own permission level.

See set permissions for details on setting user permissions for commands.

Syntax

Add a user

```
set user add id=number newname=string
```

Remove a user

```
set user remove {id=range|name=string}
```

Associate a user with a group

```
set user associate {id=number|name=string} {gid=number|gname=string}
```

Disassociate a user from a group

```
set user disassociate {id=number|name=string} {gid=number|gname=string}
```

Change user configuration attributes

```
set user [id=range|name=string]
  [newname=string]
  [commandline={on|off}]
  [groupaccess={on|off}]
  [defaultaccess={none|commandline|group}]
  [defaultgroup={none|gid|gname}]
```

Display user configuration attributes

```
set user {id=range|name=string}
```

Display user configuration attributes for all users

```
set user
```

Load an SSH public key

```
set user public_key=tftphost:filename
```

Remove an SSH public key

```
set user public_key=clear
```

Options

add

Add a user. New users are created with the default permissions (see User Models and User Permissions in Digi devices). A maximum of **32** users can be defined.

remove

Remove users.

associate

Associate a user with a group. A user can be associated with a maximum of two groups.

disassociate

Disassociate a user from a group.

id=*range*

Specifies the ID or range of IDs of the users to be acted on.

name= *string*

Specifies the name of the user to be acted on.

newname=*string*

Specifies a new user name.

gid=*number*

Specifies the identifier for the group being associated with a user. If omitted, the **gname** option must be specified.

gname=*string*

Specifies the name of the group being associated with a user. If omitted, the **gid** option must be specified.

commandline={on|off}

Specifies whether the user is allowed to access the command line of the device.

on

User can access the command line interface.

off

User can not access the command line interface.

The default is **on**.

groupaccess={on|off}

Specifies whether the user is allowed to use the access rights for any associated groups. This allows a group to define the access rights for users. For instance, if the user has **commandline=off** and an associated group has **commandline=on**, the user will have command line access if **groupaccess=on**.

on

The user can use group access rights.

off

The user cannot use group access rights.

The default is **off**.

defaultaccess={none|commandline|group}

Specifies the default access method and interface that a user will be given upon logging into the device. Note that the specified interface must be enabled for the user and have a valid menu and/or group if specified.

none

The user has no default access to the device and must explicitly specify the access type. If the user and/or associated group has no access rights then the user is not allowed to access either the command line interface or the custom menu interface.

commandline

The user displays and given access to the command line interface assuming the user and/or associated groups have command line access rights enabled.

group

The user displays the default access interface as specified by the **defaultgroup** option, assuming the specified group is valid and associated to this user. This allows the default access for a user to be controlled by the associated group.

The default is **commandline**.

defaultgroup={none|gid|gname}

Specifies the default group to use when checking the default access rights when the **defaultaccess** option is set to **group**. The specified group must be valid and associated to the user.

none

The user will not be given any default access.

gid

Group ID. The user will be given the default access method according to the default access of the group with the specified group ID.

gname

Group name. The user will be given the default access method according to the default access of the group with the specified group name.

The default is **none**.

public_key={*tftphost*:*filename*|clear}

Loads or clears an SSH public key used for authentication of this user. The key must be an RSA public key in either OpenSSH or the IETF draft format.

tftphost:filename

Loads an SSH2 public key for use with this user, where:

tftphost

The IP address or DNS name of a host from which the SSH public key will be downloaded to the Digi device using TFTP.

filename

The name of a file on the host that contains the SSH public key. If your host's implementation requires a complete path to this file, specify the path here as well.

clear

Unloads an SSH public key.

Examples

Add a new user

```
#> set user add newname=jsmith id=4
```

Remove user 7

```
#> set user remove id=7
```

Associate user "johndoe" with the root group

```
#> set user associate name=johndoe gname=root
```

Disassociate user 15 from group 2

```
#> set user disassociate id=15 gid=2
```

Set a new user name to be entered at login

```
#> set user id=4 newname=jdoe
```

Set a user to have default command line interface access

```
#> set user id=4 commandline=on defaultaccess=commandline
```

Set a user to use group access rights

```
#> set user name=johndoe groupaccess=on defaultaccess=group defaultgroup=root
```

See also

- User Models and User Permissions in Digi devices
- newpass
- revert: The **revert auth** command reverts the settings configured by set user.
- set group
- set login
- set menu
- set permissions
- show: The **show user** command shows the current user settings in a Digi device.

set video

Purpose

Configures or displays video settings for ConnectPort Display.

Syntax

Configure video settings

```
set video mode={640x480@60-16|800x600@56-16|800x600@60-16|
1024x768@70-8}
splash_time=0-30 seconds
```

Display current video settings

```
set video
```

Options

**mode={640x480@60-16|800x600@56-16|800x600@60-16|
1024x768@70-8}**

The resolution, refresh rate, and color depth of the display screen.

splash_time=0-30 seconds

The amount of time, in seconds, to show the splash screen. Valid values are 0 through 30. A value of **0** disables the splash screen.

Example

```
#> set video mode=800x600@60-16 splash_time=5
```

See also

- revert: The **revert video** command reverts the settings configured by this command.
- set putty
- set vncclient
- show: The **show video** command shows the current video settings in a Digi device.

set vncclient

Purpose

Configures or displays remote-access settings for a ConnectPort Display. ConnectPort Display can provide remote access to a computer on the network or Internet using the VNC (Virtual Network Computing) protocol.

VNC server software must be installed on the remote computer. A VNC server is provided on ConnectPort Display Software and Documentation CD.

Interaction with the remote computer is possible using a keyboard and mouse connected to the USB ports on your ConnectPort Display.

Syntax

Configure remote-access settings

```
set vncclient [state={on|off}]
  [server=vnc server ipaddr|dns name]
  [port=vnc server network port]
  [password=vnc server password]
  [reconnect=0-2000000 seconds]
  [shared={on|off}]
  [localcursor={on|off}]
  [keepalive={on|off}]
```

Display current remote-access settings

```
set vncclient
```

Options

state={on|off}

Enables or disables the connection to a remote computer's VNC server.

server={*vnc server ipaddr*|*dns name*}

The VNC server's IP address or DNS name.

port=*vnc server network port*

The network port number to connect to on the VNC server. The default port number for VNC servers is **5900**.

password=*vnc server password*

The password for logging on to the VNC server.

reconnect=0-2000000 seconds

The maximum amount of time to wait before attempting to reconnect to the VNC server if the connection cannot be established or lost.

shared={on|off}

Specifies whether the VNC server desktop can be shared with other clients. If **shared=on**, other VNC clients can connect to the VNC server while your ConnectPort Display is connected.

localcursor={on|off}

Enables or disables local mouse cursor handling. Tracking the mouse cursor locally can improve mouse performance, especially with a slow VNC server or slow network.

keepalive={on|off}

Indicates whether TCP keep-alives will be sent while connected to the VNC server. Keep-alives help to detect when a connection has been lost. TCP keep-alive parameters (such as how often to send them) are configured globally.

Example

```
#> set vncclient state=on server=10.20.1.107 port=5900 password=dnf10
reconnect=10 shared=onscalefactor=1 localcursor=on
```

See also

- revert: The **revert vncclient** command reverts the settings configured by this command.
- set service. The VNC Client Listen Daemon and VNC Server services are enabled and disabled by the **set service** command.
- set video
- show: The **show vncclient** command shows the current remote access settings in a ConnectPort Display device.
- The *ConnectPort Display User Guide* section titled **Configure Remote Access (VNC Client) Settings**.

set vpn

Purpose

Configures Virtual Private Network (VPN) settings.

Using the web interface to configure VPN settings is recommended instead of this command as it is generally easier to configure settings through that interface. Go to **Configuration > Network > Virtual Private Network (VPN) Settings**.

If you need to configure the VPN settings using scripts, use the **set vpn** command.

Virtual Private Networks (VPN) are used to securely connect two private networks together so that devices may connect from one network to the other network using secure channels. VPN uses IP Security (IPSec) technology to protect the transferring of data over the Internet Protocol (IP). The Digi device is responsible for handling the routing between networks. Devices within the private network of the Digi device can connect directly to devices on the other private network to which the VPN tunnel is established. The VPN tunnels are configured using various security settings and methods to ensure the networks are secured.

There are several uses of the **set vpn** command:

- To configure global VPN options.
- To configure use of antireplay.
- To configure support for Dynamic DNS.
- To configure support for remote peers which implement obsolete versions of NAT-T.
- To configure support for remote peers which do not fully implement the IPSEC RFCs.
- To configure and modify VPN tunnel options. VPN Tunnels define the actual tunnels that exist between two private networks. The tunnels specify the information required to establish the secure channel, the routing between the networks, and the security policies used to encrypt and authorize the data. Connect WAN products support up to two VPN tunnels. ConnectPort X products support up to five VPN tunnels.
- Configuring a VPN tunnel requires the remote VPN endpoint and the method to establish the VPN tunnel. These settings are typically specified by the remote VPN server and should correspond accordingly. Both manually keyed and ISAKMP tunnels can be configured.
- To configure IKE/ISAKMP SA Phase 1 and Phase 2 options, which create an authenticated secure channel and specify how IKE negotiates security associations (SAs).
- To display current VPN settings.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-vpn=read** to display settings, and **set permissions s-vpn=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Basic syntax

The basic syntax for **set vpn** is:

```
set vpn [options...]
```

Where ***options*** are keywords that identify groups of VPN options:

```
[global] [tunnel] [phase1] [phase2] [interface]
```

Syntax for each group of settings follow.

Configure VPN global options

```
set vpn global [options...]
```

Where ***options*** are:

```
    antireplay={on|off}
    suppress_phase1_lifetimes={on|off}
    suppress_delete_sa_for_pfs={on|off}
    send_natt_draft_01_id={on|off}
    send_natt_draft_02_id={on|off}
    send_natt_draft_03_id={on|off}
    dynamic_dns={on|off}
```

Configure VPN tunnels

```
set vpn tunnel [options...] [manually-keyed options...]|
  [isakmp options]..]
```

Where ***options*** are:

```
    tunnel=1-5
    name=tunnel name
    newname=tunnel name
    mode={disabled|manually-keyed|isakmp}
    autostart={disabled|enabled}
    host_mode={disabled|enabled}
    host_mode_security={disabled|enabled}
    host_address=ip address
    interface={eth0|mobile0}
    remote_peer_address={fqdn|ip address}
    remote_tunnel_addr=ip address
    remote_tunnel_mask=subnet mask
    remote_tunnel_range=ip address-ip address
    local_tunnel_addr=ip address
    local_tunnel_mask=subnet mask
```

Where ***manually-keyed options*** (**mode=manually-keyed**) are:

```
    inbound_spi=256-2^32
    inbound_authentication={none|md5|sha1}
    inbound_auth_key={ascii key|hex key}
```

```
inbound_encryption={none|des|3des|aes}
inbound_enc_key=ascii key|hex key
outbound_spi=256-2^32
outbound_authentication={none|md5|sha1}
outbound_auth_key=(ascii key|hex key)
outbound_encryption={none|des|3des|aes}
outbound_enc_key={ascii key|hex key}
```

and ***isakmp options*** (**mode=isakmp**) are:

```
remote_peer_id={fqdn|ip address|username}
shared_key={ascii key|hex key}
aggressive_mode={disabled|enabled}
natt_enable={disabled|enabled}
natt_ka_interval=5-255
pfs={disabled|enabled}
dh_group_phase2={1|2|5|14}
```

Note For proposals, see syntax for Show VPN IKE/ISAKMP SA Phase 1 options for tunnels (**set vpn phase2**).

Display VPN tunnel configuration settings

See syntax for Show VPN IKE/ISAKMP SA Phase 1 options for tunnels.

Set IKE/ISAKMP SA Phase 1 Options

```
set vpn phase1 [options...]
```

Where ***options*** are:

```
tunnel=1-5
name=tunnel name
proposal=1-2
state={disabled|enabled}
auth_method={shared_key|dsa_sig|rsa_sig}
authentication={md5|sha1}
encryption={des|3des|aes}
encryption_size={0|128|192|256} (bits)
sa_lifetime=10-2^32 (seconds)
sa_lifetime_data=0-2^32 (kilobytes)
diffie_hellman_group={1|2|5|14}
```

Display IKE/ISAKMP SA Phase 1 Options

See Show VPN IKE/ISAKMP SA Phase 1 options for tunnels.

Set IKE/ISAKMP SA Phase 2 Options

```
set vpn phase2 [options...]
```

Where ***options*** are:

```
tunnel=1-5
name=tunnel name
proposal=1-8
state={disabled|enabled}
```

```
    authentication={none|md5|sha1}
    encryption={none|des|3des|aes}
    encryption_key_length={0|128|192|256] (0=use default key length)
     sa_lifetime=60-2^32 (seconds)
     sa_lifetime_data=0-2^32) (kilobytes)
```

Display IKE/ISAKMP SA Phase 2 Options

See Show VPN IKE/ISAKMP SA Phase 2 options for tunnels.

Select the network interface used to communicate with the remote VPN device

```
set vpn interface [interface={eth0|mobile0}]
  [local_peer_id={fqdn|interface address|username|
  certificate dn}]
```

Display the network interface used to communicate with the remote VPN device

See Show the network interface used to communicate with the remote VPN device.

Options

VPN global options

set vpn global

Specifies that the **set vpn** command is for setting global VPN options.

antireplay={on|off}

Specifies whether the antireplay feature is on or off. Antireplay allows the IPsec tunnel receiver to detect and reject packets that have been replayed. It does this by adding information to the packets exchanged between VPN endpoints, to ensure that a third party cannot replay the same information to one of the VPN endpoints at a later time to recreate the secure channel again.

CAUTION! If you are using manually-keyed tunnels, disable this option.

For negotiations to succeed, both the local and remote sides of the connection must be set to the same value. Set this field to match that at the remote VPN gateway. The default is **on**.

suppress_phase1_lifetimes={on|off}

Specifies whether ISAKMP phase 1 lifetimes should be suppressed. Some VPN equipment does not negotiate the ISAKMP phase 1 lifetimes. This equipment may refuse to negotiate with this unit if it includes lifetime values in the phase 1 negotiation messages. Set this option to **on** to prevent the phase 1 lifetimes from being included in the ISAKMP phase 1 messages if this unit must communicate with this type of equipment. However, in most cases, this option should be set to **off**.

suppress_delete_sa_for_pfs={on|off}

Specifies whether delete notifications for any phase 2 security associations (SAs) are suppressed. In most cases this option should be set to **off**. VPN devices usually send a delete notification for any phase 2 SAs that are left over from previous sessions when they start to negotiate quick mode. However, some devices do not handle this notification correctly and will terminate the connection

when they receive it. If you have trouble connecting to the remote VPN device, try setting this option to **on** to suppress sending this message.

send_natt_draft_01_id={on|off}

Use this option to control whether the unit should support draft 01 of the NAT-T protocol. This is an obsolete version of the protocol and support for it should only be enabled if the remote peer requires it

send_natt_draft_02_id={on|off}

Use this option to control whether the unit should support draft 02 of the NAT-T protocol. This is an obsolete version of the protocol and support for it should only be enabled if the remote peer requires it.

send_natt_draft_03_id={on|off}

Use this option to control whether the unit should support draft 03 of the NAT-T protocol. This is an obsolete version of the protocol and support for it should only be enabled if the remote peer requires it.

dynamic_dns={on|off}

Specifies whether the IP addresses of remote VPN peers may change on the fly, known as dynamic DNS. Set to **on** if you are specifying the address of the remote VPN device with a DNS name, and that device uses dynamic DNS because its public IP address can change. This causes the Digi device to poll the DNS server once a minute to see if the remote VPN device's IP address has changed. The IPSec software will be restarted with the new IP address if it does change.

Setting this option to **on** increases network traffic, since the unit polls the DNS server once a minute.

This example demonstrates how to set the global configuration settings to enable anti-replay and dynamic DNS:

```
set vpn global antireplay=on
set vpn global suppress_phase1_lifetimes=off
set vpn global suppress_delete_sa_for_pfs=off
set vpn global send_natt_draft_01_id=off
set vpn global send_natt_draft_02_id=off
set vpn global send_natt_draft_03_id=off
set vpn global dynamic_dns=on
```

VPN tunnel configuration options

set vpn tunnel

Specifies that the **set vpn** command is for configuring a VPN tunnel.

options

The VPN tunnel configuration options. The set of options specified depends on whether the method of establishing the VPN tunnel is manually-keyed or ISAKMP.

tunnel=1-5 (for ConnectPort X products)
tunnel=1-2 (for Connect WAN products)

The index number for a new or existing VPN tunnel.

name=*tunnel name*

A name that describes the VPN tunnel. This may be used to help identify each tunnel with a descriptive and unique name.

newname=*tunnel name*

The new name for the VPN tunnel.

mode={disabled|manually-keyed|isakmp}

The method of establishing the VPN tunnel.

disabled

The VPN tunnel is enabled or disabled. Use this option when creating several tunnels, where only one would be used initially. In that case, you would add a disabled tunnel for future use and enable it on a subsequent **set vpn** command.

manually-keyed

You should only use this option if the remote peer does not support IKE/ISAKMP. The VPN tunnel is established by manually keying in VPN tunnel and security settings. These settings must match the settings of the remote VPN endpoint. Manually-keyed VPNs do not use IKE/ISAKMP. Manually-keyed VPN tunnels are less secure than tunnels secured with IKE/ISAKMP because the encryption keys never expire, and so the same encryption keys are always used.

isakmp

This is the preferred mode of operation. In this mode, a set of security policies which are used to negotiate a secure connection to the remote VPN peer. When the tunnel is brought up, IKE/ISAKMP is used to negotiate a fresh set of encryption keys. If the tunnel is used for a long time, then a new set of keys is renegotiated periodically. Since the keys are replaced every time the tunnel is brought up, and then periodically afterwards, it is much more secure.

autostart={disabled|enabled}

Specifies whether to negotiate the VPN tunnel as soon as the network interface used for it comes up. Set to **enabled** if the Digi device should establish the VPN tunnel as soon as the network interface selected is ready to use. Set to **disabled** if the Digi device should wait until a device on the local private network attempts to communicate with a device on the remote network before establishing the VPN tunnel.

host_mode={disabled|enabled}

This option determines whether the local side of the VPN tunnel will be visible to users on the remote peer. Enable host mode to hide the addresses of devices on the local side of the VPN tunnel from the remote side. In this case, devices on the remote side only sees a single IP address which you set with the **host_address** option below. Disable this option to allow the remote side to see the local subnet which is the local end of the VPN tunnel.

If this option is enable, the **set nat** command must be used to enable NAT on the VPN interface associated with this tunnel. The VPN interfaces listed by NAT are zero based, so VPN tunnel **1** is associated with interface **VPN0**, and so on. For more information, see set nat.

host_mode_security={disabled|enabled}

This is an optional feature that you can leave disabled. If this option is enabled it, IPSec discards any traffic from the local side of the VPN tunnel which is not from the subnet specified by the **local_tunnel_addr**, **local_tunnel_mask**, and **local_tunnel_range** options below.

host_address=*ip address*

Use this option to set the IP address visible to devices on the remote end of the VPN tunnel when **host_mode** is enabled.

interface={eth0|mobile0}

The network interface that is used as the local endpoint of the VPN tunnel. This interface communicates with the remote VPN peer. The identity set for this interface with the **set vpn interface** command is the one sent to the remote VPN peer during the ISAKMP negotiation.

eth0

Ethernet network interface.

mobile0

Mobile network interface a **mobile0** device has a cellular modem. In most cases, this is the correct device to use to communicate with a remote VPN device on the Internet.

remote_peer_address={*fqdn*|*ip address*}

The IP address or hostname of the peer with which the VPN connection is established.

remote_tunnel_addr=*ip address*
remote_tunnel_mask=*subnet mask*
remote_tunnel_range=*ip address-ip address*

These options specify the range of IP addresses on the remote side of the tunnel. Traffic addressed to these IP addresses from the local side of the tunnel will be sent through the tunnel to the remote network. The remote VPN peer sends traffic from these addresses through the tunnel to the local side.

Digi devices support a mode of VPN tunnel operation called *VPN tunnel all mode*, where all traffic that is not directed to the local subnet is sent across a VPN tunnel to a remote network. This mode is different from the normal mode of VPN tunnel operation, where the range of the remote subnet is set explicitly. VPN tunnel all mode is supported when the Digi device is the initiator of the VPN connection. It is not supported when the Digi device is the server.

For example, in the normal mode of operation, a user might set up a VPN tunnel between the local subnet at **192.168.1.0/24** to a remote subnet at **172.16.1.0/24**. In this case, the remote subnet range is the subnet at **172.16.1.x. In** VPN tunnel all mode, the remote subnet is any address that is not on the local subnet, or in this case, anything not in the subnet **192.16.1.x**.

The local subnet must be defined as a specific range, for example **192.168.1.0/24**. This is specified in the VPN settings by setting the IP address of the local subnet to **192.168.1.0**, and the subnet mask to **255.255.255.0**. VPN tunnel all mode is specified by setting the remote IP address to **0.0.0.0**, and the remote subnet mask to **0.0.0.0**.

With the configuration described above, any frames sent from the **192.168.1.x** network to any IP address not in the **192.168.1.x** subnet will be set over the VPN tunnel to the remote subnet.

When configuring a Digi device for VPN tunnel all mode and the device allows for setting the gateway priority, set the gateway priority. The gateway priority is set by the **set network gwpriority**" option (see "set network" on page 288). Set **gwpriority** to **eth0** for Ethernet-enabled Digi device, or to **wln0** for wireless Digi devices.

If the Digi device's IP address on the Ethernet (or wireless) interface is statically configured, specify the address for the gateway on that interface. The gateway IP address is set by the **set network** command.

local_tunnel_addr=*ip address*
local_tunnel_mask=*subnet mask*
local_tunnel_range=*ip address-ip address*

If host_mode is disabled, these options specify the range of IP addresses at the local side of the VPN tunnel. Traffic from devices in this range on the remote side of the tunnel will be tunneled to the other side of the tunnel. Devices at the remote side of the tunnel will be able to send frames to IP addresses within the subnet.

If **host_mode** and **host_mode_security** are both enabled, these options specify the range of IP addresses that are allowed to communicate with devices on the remote side of the tunnel.

***manually-keyed options* (mode=manually-keyed):**

These options are for VPN manually-keyed VPN tunnels. To properly configure a manual-keyed tunnel, the following settings are required to be set as specified by the remote VPN server. This includes the

local and remote network settings that handle the routing between the local and remote peers. It also includes the security settings for both incoming and outgoing traffic, which may be different from each other, depending on the implementation of the remote VPN server. Incoming or inbound traffic is defined as any traffic sent from a remote peer on the remote network of the remote VPN endpoint to a local peer on the local network. Outgoing or outbound traffic is defined as any traffic sent from a local peer to a remote peer.

inbound_spi=256-2^32

The Security Parameter Index (SPI) for inbound traffic. The SPI defines the unique index for a tunnel used to identify the security settings for IPSec. The SPI is a 32-bit unsigned value that must not be less than 256.

inbound_authentication={none|md5|sha1}

The optional authentication algorithm, used with the associated authentication key specified by the **inbound_auth_key** option, to authorize access on the VPN tunnel for inbound traffic.

none

No authentication algorithm is used.

md5

MD5 authentication algorithm, which uses 128-bit keys.

sha1

SHA1 authentication algorithm, which uses 160-bit keys.

inbound_auth_key={*ascii key*|*hex key*}

The authentication key for inbound traffic, according to the authentication algorithm specified by the **inbound_authentication** option. The authentication key may be specified as an ASCII value using alpha-numeric characters or may be specified as a hexadecimal value prefixed by **0x**. The following table lists the associated lengths of the authentication keys based on the authentication algorithm.

Algorithm	Size	Key Length	
		ASCII	Hexadecimal
MD5	128-bit	16	32
SHA1	160-bit	20	40

inbound_encryption={none|des|3des|aes}

The optional encryption algorithm used with the associated encryption key specified by the **inbound_enc_key** option to encrypt data on the VPN tunnel for inbound traffic.

none

No encryption algorithm is used.

des

DES encryption algorithm, which uses 64-bit keys.

3des

3DES encryption algorithm, which uses 192-bit keys.

aes

AES encryption algorithm, which uses 128-bit keys.

inbound_enc_key={*ascii key*|*hex key*}

The encryption key for inbound traffic, according to the authentication algorithm specified by the **inbound_encryption** option. The encryption key may be specified as an ASCII value using alpha-numeric characters or may be specified as a hexadecimal value prefixed by **0x**. The following table lists the associated lengths of the encryption keys based on the encryption algorithm.

Algorithm	Size	Key Length	
		ASCII	Hexadecimal
DES	64-bit	8	16
3 DES	192-bit	24	48
AES	128-bit	16	32

outbound_spi=256 - 2^32

The SPI for outbound traffic.The SPI defines the unique index for a tunnel used to identify the security settings for IPSec. The SPI is a 32-bit unsigned value that must not be less than 256.

outbound_authentication={none|md5|sha1}

The optional authentication algorithm used with the associated authentication key specified by the **outbound_auth_key** option to authorize access on the VPN tunnel for outbound traffic.

none

No authentication algorithm is used.

md5

MD5 authentication algorithm, which uses 128-bit keys.

sha1

SHA1 authentication algorithm, which uses 160-bit keys.

outbound_auth_key={*ascii key*|*hex key*}

The authentication key for outbound traffic, according to the authentication algorithm specified by the **outbound_authentication** option. The authentication key may be specified as an ASCII value using alpha-numeric characters or may be specified as a hexadecimal value prefixed by **0x**. For the allowed lengths for this key, see **inbound_auth_key**.

outbound_encryption={none|des|3des|aes}

The optional encryption algorithm used with the associated encryption key specified by the **outbound_enc_key** option to encrypt data on the VPN tunnel for outbound traffic. For the allowed values, see **inbound_encryption**.

outbound_enc_key={*ascii key*|*hex key*}

The encryption key for outbound traffic, according to the authentication algorithm specified by the **outbound_encryption** option. For the allowed values and key length, see **inbound_enc_key**.

***isakmp options* (mode=isakmp)**

To configure an ISAKMP tunnel, you must configure the settings to match those on the remote VPN server.

To specify security proposals for VPN ISAKAMP tunnels, see Set IKE/ISAKMP SA Phase 1 Options.

remote_peer_id={*fqdn*|*ip address* | *username*}

The IP address or hostname of the peer with which the VPN connection is established.

shared_key={*ascii key*|*hex key*}

A key that secures the VPN tunnel. The key can be either an ASCII value using alphanumeric characters or a hexadecimal value prefixed by **0x**.

aggressive_mode={enabled|disabled}

Enables or disables aggressive mode for negotiating Internet Key Exchange (IKE) Phase One using Internet Security Association and Key Management Protocol (ISAKMP). Negotiations establish security settings and a secure channel for subsequent messages. For the negotiations to progress, both sides must be configured identically. Aggressive mode processes Phase One negotiations using fewer exchanges than Main Mode processing. In the first exchange, almost everything is sent in the proposed IKE values, including the Diffie-Hellman key, nonce to sign and verify, and the identity. The weakness of using Aggressive Mode compared to Main Mode is that negotiations exchange information before the secure channel is created. However, because fewer exchanges are used, aggressive mode is faster than main mode. Aggressive mode may be required when a peer gateway IP address is dynamic.

If **aggressive_mode** is disabled, Main Mode processing is used.

natt_enable={disabled|enabled}

Set this option to enabled if there is a firewall between the two VPN peers. Enabling this option will cause the unit to negotiate a NAT Traversal connection which maintains the VPN tunnel through firewalls. Both VPN peers need to be configured the same way.

Enabling this option generates additional traffic.

natt_ka_interval=5-255

Use this option to set the NAT-T keep alive interval. The interval is specified in seconds and determines how often the unit will send NAT keep alive frames to prevent the NAT firewall from timing out the connection. This value should be set to less than half the of the timeout value used by the NAT firewall.

pfs={enabled|disabled}

Specifies whether the Perfect Forward Secrecy (PFS) method is on or off. PFS is a method of deriving session keys from known keying material. PFS establishes greater resistance to cryptographic attacks by ensuring that a given key of an IKE SA is not derived from any other secret, and that no other key can be derived from this key.

For negotiations to succeed, both the local and remote sides of the connection must have the **pfs** and **dh_group** options set to the same values.

The default is **on**.

dh_group_phase2={1|2|5|14}

The Diffie-Hellman (DH) prime modulus group. Diffie-Hellman is a public-key cryptography protocol for establishing a shared secret over an insecure communications channel. Diffie-Hellman is used with IKE to establish the session keys that create a secure channel. This setting is used if Perfect Forward Secrecy is also enabled (**pfs=on**.)

Digi Cellular Family products support the following Diffie-Hellman prime modulus groups:

1

Group 1 (768-bit).

2

Group 2 (1024-bit).

5

Group 5 (1536-bit).

14

Group 14 (2048-bit).

The default is **2** (Group 2).

About IKE/ISAKMP SA Phase 1 and Phase 2 options

Internet Key Exchange (IKE) negotiates the IPSec security associations (SA). This process requires that the IPSec systems first authenticate themselves to each other and establish ISAKMP (IKE) shared keys. The SAs are relationships between two or more entities or peers that describe how the entities or peers uses security services to communicate securely.

IKE negotiations are handled using two different phases.

- Phase 1 is responsible for creating an authenticated and secure channel between the two peers. Typically, phase one is completed using a Diffie-Hellman exchange using cryptography.
- Phase 2 is responsible for negotiating the final SAs and generating the required keys and key material for IPSec. This is completed by negotiating one or more sets of security policies, or proposals, between the two peers until a given set is agreed upon by both peers.

IKE/ISAKMP SA Phase 1 options

The options below allow you to specify the phase 1 proposals which are used during the first phase of the ISAKMP negotiation. Each proposal specifies a set of security parameters which are to be used to create the phase 1 connect. When the phase 1 negotiation takes place, the local and remote VPN peers compare their lists of phase 1 policies and select the strongest one they both have in common. The settings in the selected policy are used to create the phase 1 connection.

set vpn phase1

Specifies that the **set vpn** command is for configuring a VPN Phase 1 options.

options

tunnel=1-5

The index number assigned to the VPN tunnel.

name=*tunnel name*

The name of the VPN tunnel.

proposal=1-2

The index number assigned to the security proposal.

state={enabled|disabled}

Whether the phase 1 proposal is enabled or disabled.

auth_method={shared_key|dsa_sig|rsa_sig}

The authentication method performed.

shared_key

Authentication is performed by using a key that secures the VPN tunnel, where the key is either an ASCII alphanumeric value or a hexadecimal value.

dsa_sig

Authentication is performed using a DSA certificate that has been uploaded to the Digi device.

rsa_sig

Authentication is performed using an RSA certificate that has been uploaded to the Digi device.

For more information on certificate management and uploading certificates, see certmgmt.

authentication={md5|sha1}

The authentication algorithm used in IKE negotiations to authenticate the IKE peers and Security Associations (SAs).

md5

MD5 authentication algorithm, which uses 128-bit keys.

sha1

SHA1 authentication algorithm, which uses 160-bit keys.

encryption={des|3des|aes}

The encryption algorithm used in IKE negotiations for encrypting data.

des

DES encryption algorithm, which uses 64-bit keys.

3des

3DES encryption algorithm, which uses 192-bit keys.

aes

AES encryption algorithm, which uses 128-bit keys.

encryption_size={0|128|192|256} (bits)

The encryption key length, in bits, used in IKE negotiations for encrypting data. The key length is based on the encryption algorithm and is used to calculate and create the shared key.

sa_lifetime=10-2^32 (seconds)

Determines how long an Security Association (SA) policy is active, in seconds. After the IKE SA has been negotiated, the SA lifetime begins. Once the lifetime has completed, a new set of SA policies are negotiated using IKE phase 2 negotiation.

sa_lifetime_data=0-2^32 (kilobytes)

The amount of data, in bytes or kilobytes, sent and received until the SA is renegotiated. This value is analogous to the SA lifetime. Also known as **SA life size**.

diffie_hellman_group={1|2|5|14}

The Diffie-Hellman (DH) prime modulus group. Diffie-Hellman is a public-key cryptography protocol for establishing a shared secret over an insecure communications channel. Diffie-Hellman is used with IKE to establish the session keys that create a secure channel. This setting is used if Perfect Forward Secrecy is also enabled (**pfs=on**.)

Digi Cellular Family products support the following Diffie-Hellman prime modulus groups:

1

Group 1 (768-bit).

2

Group 2 (1024-bit).

5

Group 5 (1536-bit).

14

Group 14 (2048-bit).

The default is 2 (Group 2).

IKE/ISAKMP SA Phase 2 options

Security policies define the set of security settings for incoming and outgoing traffic used to encrypt and authorize data. One or more sets of settings may be specified. When the phase 2 connection is negotiated, the local and remote VPN peers compare their list of policies and select the most secure one they both have in common.

The VPN Phase 2 options are used to configure a set of security policies for ISAKMP tunnels. The settings define the set of encryption and authentication algorithms used for incoming and outgoing traffic over the VPN tunnel.

A security policy can have multiple proposals. For example, a policy can have two proposals to allow older VPN devices to connect using less-secure methods, while allowing the same policy to have a second (or more) proposal to allow newer, more powerful end-points to use more secure methods.

set vpn phase2

Specifies that the **set vpn** command is for configuring a VPN Phase 2 options.

options

tunnel=1-5

The index number assigned to the VPN tunnel.

name=*tunnel name*

The name of the VPN tunnel.

proposal=(1- 8)

The index number assigned to the security proposal.

state={enabled|disabled}

Whether the VPN tunnel is enabled or disabled. You can use this option when creating several tunnels where only one would be used initially. In that case, you would add a disabled tunnel for future use and enable it on a subsequent **set vpn** command.

authentication={none|md5|sha1}

The authentication algorithm used in authenticating clients.

none

This option is used for debugging purposes only. It is not secure and most VPN devices will not accept it.

md5

MD5 authentication, which uses 128-bit keys.

sha1

SHA1 authentication, which uses 160-bit keys.

encryption={none|des|3des|aes}

The encryption algorithm used for encrypting data. AES is generally considered to be more secure than DES, and longer keys are more secure than shorter keys. However, using longer keys may reduce throughput.

none

This option is used for debugging purposes only. It is not secure and most VPN devices will not accept it.

des

DES encryption, which uses 64-bit keys.

3des

3-DES encryption, which uses 192-bit keys.

aes

AES encryption, which uses either 128-bit, 192-bit, or 256-bit keys depending on the negotiated security settings.

encryption_key_length={0|128|192|256] (0=use default key length)

The encryption key length for AES. Set this option to **0** when using DES or 3DES to select the default key lengths. Set this option to the desired key length when using AES. Longer keys are more secure, but may reduce throughput.

sa_lifetime=60-2^32 (seconds)

Determines how long a Security Association (SA) policy is active, in seconds. After the SA has been negotiated, the SA lifetime begins. Once the lifetime has completed, a new set of SA policies are negotiated with the remote VPN endpoint. Shorter lifetimes are more secure since the encryption keys are replaced more often, however, data transfer will be paused a couple seconds every time a key negotiation takes place.

sa_lifetime_data=0-2^32) (kilobytes)

The amount of data, in bytes or kilobytes, that is sent and received until the SA is renegotiated. This value is analogous to the SA lifetime. Also known as **SA life size**.

VPN network interface options

These options set the local identity used for the ISAKMP negotiation. The unit must identify itself to the remote VPN peer during the ISAKMP negotiation. The identity can be a Fully Qualified Domain Name (FQDN), the IP address of the interface used for the negotiation, a username, or a public key certificate. The identity is associated with the network interface used to communicate with the remote VPN peer.

Use the **set vpn interface** command to set the identity for the network interface.

Syntax

```
set vpn interface [interface={eth0|mobile0}]
  [local_peer_id={fqdn|interface address|username|
  certificate dn}]
```

Options

interface={eth0|mobile0}

The network interface used to communicate with the remote VPN device.

eth0

Ethernet network interface.

mobile0

Mobile network interface (in most units this is the cellular modem). In most cases, this is the correct device to use to communicate with a remote VPN device on the Internet.

local_peer_id={*fqdn*|*interface address*|*username*|*certificate dn*}

Sets the identity associated with the network interface. This identity is passed to the remote VPN peer during the ISAKMP negotiation. This option must be set to match the configuration of the **remote peer identity** on the remote VPN peer.

Examples

Set global parameters

The following example demonstrates how to set global parameters. Antireplay and dynamic DNS support are turned on. The other global options are turned off.

```
#> set vpn global antireplay=on
```

```
#> set vpn global suppress_phase1_lifetimes=off
```

```
#> set vpn global suppress_delete_sa_for_pfs=off
```

```
#> set vpn global send_natt_draft_01_id=off
```

```
#> set vpn global send_natt_draft_02_id=off
```

```
#> set vpn global send_natt_draft_03_id=off
```

```
#> set vpn global dynamic_dns=on
```

```
#> show vpn global
```

```
Global VPN Configuration :

    antireplay                  : on
    suppress_phase1_lifetimes   : off
    suppress_delete_sa_for_pfs  : off
    send_natt_draft_01_id       : off
    send_natt_draft_02_id       : off
    send_natt_draft_03_id       : off
    dynamic_dns                 : on
```

Set peer IDs

The following example demonstrates how to set the peer ID for the **mobile0** interface to use the IP address currently assigned to that interface:

```
#> set vpn interface interface=mobile0 local_peer_id=Interface-Address
```

```
#> show vpn interface
```

```
VPN Interface Configuration :

    Interface    Local Peer Name
    ==========   ============================================
    eth0       : 00:30:9D:01:01:FE@digi.com
    mobile0    : Interface-Address
```

The following example demonstrates how to set the peer ID for the **mobile0** interface to the FQDN **localpeer.digi1.com**:

```
#> set vpn interface interface=mobile0 local_peer_id=localpeer.digi1.com
```

```
#> show vpn interface
```

```
VPN Interface Configuration :

    Interface    Local Peer Name
    ==========   ============================================
    eth0       : 00:30:9D:01:01:FE@digi.com
    mobile0    : localpeer.digi1.com
```

Configure a VPN tunnel

The following example demonstrates how to configure a VPN tunnel.

ISAKMP is used to negotiate the connection over the cell modem interface to a remote peer with the identifier FQDN **remotepeer.digi1.com**.

1. Since the **mobile0** interface was selected by an earlier command, our identifier will be the one assigned to the **mobile0** interface through the **set vpn interface** command.

```
#> set vpn tunnel tunnel=1 mode=isakmp interface=mobile0 remote_peer_
id=remotepeer.digi1.com
```

2. Set up the tunnerlas soon as the interface becomes available:

```
#> set vpn tunnel tunnel=1 autostart=enabled
```

3. Set up a standard subnet to subnet tunnel:

```
#> set vpn tunnel tunnel=1 host_mode=disabled
```

4. The remote VPN device is at the DNS address **remotepeer.digi1.com**:

```
#> set vpn tunnel tunnel=1 remote_peer_address=remotepeer.digi1.com
```

5. Set the subnet at the remote end of the tunnel:

```
#> set vpn tunnel tunnel=1 remote_tunnel_addr=192.168.1.0 remote_
tunnel_mask=255.255.255.0
```

6. Set the subnet at the local end of the tunnel:

```
#> set vpn tunnel tunnel=1 local_tunnel_addr=172.16.1.0 local_tunnel_
mask=255.255.255.0
```

7. Set the shared key used for authentication:

```
#> set vpn tunnel tunnel=1 shared_key=TheSharedKey0123456789
```

8. Enable aggressive mode:

```
#> set vpn tunnel tunnel=1 aggressive_mode=enabled
```

9. Enable NAT-T in case there is a NAT firewall between the two VPN peers:

```
#> set vpn tunnel tunnel=1 natt_enable=enabled
```

10. Set the NAT-T keep alive interval to **20** seconds:

```
#> set vpn tunnel tunnel=1 natt_ka_interval=20
```

11. Enable Perfect Forward Secrecy:

```
#> set vpn tunnel tunnel=1 pfs=enabled
```

12. Use Diffie-Hellman group **2** for the phase 2 PFS negotiation:

```
#> set vpn tunnel tunnel=1 dh_group_phase2=2
```

13. Disable proposal **1** while we set it up so we do not get error messages:

```
#> set vpn phase1 tunnel=1 proposal=1 state=disabled
```

14. Use a **shared key** to authenticate with the remote peer:

```
#> set vpn phase1 tunnel=1 proposal=1 auth_method=shared_key
```

15. Use **MD5** to authenticate individual frames:

```
#> set vpn phase1 tunnel=1 proposal=1 authentication=md5
```

16. Use **Triple DES** to encrypt phase 1 frames:

```
#> set vpn phase1 tunnel=1 proposal=1 encryption=3des
```

17. Use the **default key size** for triple DES:

```
#> set vpn phase1 tunnel=1 proposal=1 encryption_size=0
```

18. Renegotiate the phase **1** SA at least once every **8** hours:

```
#> set vpn phase1 tunnel=1 proposal=1 sa_lifetime=28800
```

19. Renegotiate the phase **1** SA whenever **50** Megabytes of data have been sent across it:

```
#> set vpn phase1 tunnel=1 proposal=1 sa_lifetime_data=50000
```

20. Use Diffie-Hellman group **2** for phase 1 PFS:

```
#> set vpn phase1 tunnel=1 proposal=1 diffie_hellman_group=2
```

21. Now this proposal can be enabled:

```
#> set vpn phase1 tunnel=1 proposal=1 state=enabled
```

22. Disable the phase **2** proposal so it can be configured:

```
#> set vpn phase2 tunnel=1 proposal=1 state=disabled
```

23. Use **MD5** to authenticate frames:

```
#> set vpn phase2 tunnel=1 proposal=1 authentication=md5
```

24. Use **triple DES** to encrypt data:

```
#> set vpn phase2 tunnel=1 proposal=1 encryption=3des
```

25. Use the **default key size**:

```
#> set vpn phase2 tunnel=1 proposal=1 encryption_key_length=0
```

26. Renegotiate keys at least once every **8** hours:

```
#> set vpn phase2 tunnel=1 proposal=1 sa_lifetime=28800
```

27. Renegotiate keys whenever **50** Megabytes of data have been transferred:

```
#> set vpn phase2 tunnel=1 proposal=1 sa_lifetime_data=50000
```

28. Now that the proposal is set up, enable it:

```
#> set vpn phase2 tunnel=1 proposal=1 state=enabled
```

29. Print out the tunnel configuration:

```
#> show vpn tunnel tunnel=1 verbose=on
```

```
VPN Tunnel #1 Configuration :
General Settings :

       name                : Tunnel 1
       mode                : isakmp
       autostart           : enabled
       host mode           : disabled
       remote peer address : remotepeer.digi1.com
       remote peer ID      : remotepeer.digi1.com
       interface           : mobile0
       local peer ID       : localpeer.digi1.com
Tunnel Settings :

       remote side         : ipv4subnet 192.168.1.0 - 255.255.255.0
       local side          : ipv4subnet 172.16.1.0 - 255.255.255.0
```

```
 ISAKMP Settings:

        Client              : enabled
        Server              : enabled
        NAT Traversal       : enabled
        NAT-T KA Interval   : 20
        Aggressive mode     : enabled
        PFS                 : enabled
        Phase 1 DH Group    : set in each phase 1 proposal
        Phase 2 DH Group    : 2 (1024-bit)
```

```
    ISAKMP Phase 1 Settings:
       index#  encryption/size  authentication
       ------  ---------------  --------------
       1       3des/0           md5
```

```
    Phase 2 Settings :

       index#  state     encryption  authentication
       ------  --------  ----------  --------------
       1       enabled   3des        md5
       2       disabled  des         md5
```

```
        3       disabled  des           md5
        4       disabled  des           md5
        5       disabled  des           md5
        6       disabled  des           md5
        7       disabled  des           md5
        8       disabled  des           md5
```

Configure a more complex VPN tunnel

This example is more complex. This script sets up a second tunnel to connect to a different VPN peer with the IP address **166.65.20.35**. The remote uses the FQDN anotherpeer.digi1.com as its identifier. Two proposals are set up for both phase 1 and for phase 2. The phase 1 proposals both use the SHA1 authentication hash. One proposal supports triple DES, the other 256-bit AES. Both phase 2 proposals specify MD5 authentication and AES encryption. One proposal specifies 256-bit keys, the other 128-bit keys.

```
#> set vpn tunnel tunnel=2 mode=isakmp interface=mobile0 remote_peer_
id=anotherpeer.digi1.com
#> set vpn tunnel tunnel=2 autostart=enabled
#> set vpn tunnel tunnel=2 host_mode=disabled
#> set vpn tunnel tunnel=2 remote_peer_address=166.65.20.35
#> set vpn tunnel tunnel=2 remote_tunnel_addr=192.168.10.0 remote_tunnel_
mask=255.255.255.0
#> set vpn tunnel tunnel=2 local_tunnel_addr=172.16.1.0 local_tunnel_
mask=255.255.255.0
#> set vpn tunnel tunnel=2 aggressive_mode=disabled
#> set vpn tunnel tunnel=2 natt_enable=enabled
#> set vpn tunnel tunnel=2 natt_ka_interval=20
#> set vpn tunnel tunnel=2 pfs=enabled
#> set vpn tunnel tunnel=2 dh_group_phase2=2
#> set vpn phase1 tunnel=2 proposal=1 state=disabled
#> set vpn phase1 tunnel=2 proposal=1 auth_method=shared_key
#> set vpn phase1 tunnel=2 proposal=1 authentication=sha1
#> set vpn phase1 tunnel=2 proposal=1 encryption=3des
#> set vpn phase1 tunnel=2 proposal=1 encryption_size=0
#> set vpn phase1 tunnel=2 proposal=1 sa_lifetime=28800
#> set vpn phase1 tunnel=2 proposal=1 sa_lifetime_data=50000
#> set vpn phase1 tunnel=2 proposal=1 diffie_hellman_group=2
#> set vpn phase1 tunnel=2 proposal=1 state=enabled
#> set vpn phase1 tunnel=2 proposal=2 state=disabled
#> set vpn phase1 tunnel=2 proposal=2 auth_method=shared_key
#> set vpn phase1 tunnel=2 proposal=2 authentication=sha1
#> set vpn phase1 tunnel=2 proposal=2 encryption=AES
#> set vpn phase1 tunnel=2 proposal=2 encryption_size=256
#> set vpn phase1 tunnel=2 proposal=2 sa_lifetime=28800
#> set vpn phase1 tunnel=2 proposal=2 sa_lifetime_data=50000
#> set vpn phase1 tunnel=2 proposal=2 diffie_hellman_group=2
#> set vpn phase1 tunnel=2 proposal=2 state=enabled
#> set vpn phase2 tunnel=2 proposal=1 state=disabled
#> set vpn phase2 tunnel=2 proposal=1 authentication=md5
#> set vpn phase2 tunnel=2 proposal=1 encryption=AES
#> set vpn phase2 tunnel=2 proposal=1 encryption_key_length=128
#> set vpn phase2 tunnel=2 proposal=1 sa_lifetime=28800
#> set vpn phase2 tunnel=2 proposal=1 sa_lifetime_data=50000
#> set vpn phase2 tunnel=2 proposal=1 state=enabled
#> set vpn phase2 tunnel=2 proposal=2 state=disabled
#> set vpn phase2 tunnel=2 proposal=2 authentication=md5
```

```
#> set vpn phase2 tunnel=2 proposal=2 encryption=AES
#> set vpn phase2 tunnel=2 proposal=2 encryption_key_length=256
#> set vpn phase2 tunnel=2 proposal=2 sa_lifetime=28800
#> set vpn phase2 tunnel=2 proposal=2 sa_lifetime_data=50000
#> set vpn phase2 tunnel=2 proposal=2 state=enabled
#> show vpn tunnel tunnel=2 verbose=on
```

```
VPN Tunnel #2 Configuration :

   General Settings :

      name                : Tunnel 2
      mode                : isakmp
      autostart           : enabled
      host mode           : disabled
      remote peer address : 166.65.20.35
      remote peer ID      : anotherpeer.digi1.com
      interface           : mobile0
      local peer ID       : localpeer.digi1.com

   Tunnel Settings :

      remote side         : ipv4subnet 192.168.10.0 - 255.255.255.0
      local side          : ipv4subnet 172.16.1.0 - 255.255.255.0


   ISAKMP Settings:

       Client              : enabled
       Server              : enabled
       NAT Traversal       : enabled
       NAT-T KA Interval   : 20
       Aggressive mode     : disabled
       PFS                 : enabled
       Phase 1 DH Group    : set in each phase 1 proposal
       Phase 2 DH Group    : 2 (1024-bit)

   ISAKMP Phase 1 Settings:

      index#  encryption/size  authentication
      ------  ---------------  --------------
      1       3des/0           sha1
      2       aes/256          sha1

   Phase 2 Settings :

      index#  state     encryption  authentication
      ------  --------  ----------  --------------
      1       enabled   aes         md5
      2       enabled   aes         md5
      3       disabled  des         md5
      4       disabled  des         md5
      5       disabled  des         md5
      6       disabled  des         md5
      7       disabled  des         md5
      8       disabled  des         md5
```

Configure a VPN tunnel with RSA certificate authentication and host mode

This example demonstrates how to set up a third tunnel to authenticate using an RSA certificate, and how to configure host mode. The VPN interface created for host mode is given the IP address **50.1.1.1**. This is the address which is visible to devices on the remote side of the tunnel. Host mode security is turned on and configured to only allow devices in the **172.16.1.0** local subnet to communicate over the host mode connection. The NAT firewall will also have to be configured to support host mode. [Get an example of how to do this from Mark Weber]

In addition to the entering the configuration commands below, you would have to send updates for the appropriate RSA certificates to the Digi device.

```
#> set vpn tunnel tunnel=3 mode=isakmp interface=mobile0
remote_peer_id=Certificate-DN
#> set vpn tunnel tunnel=3 autostart=enabled
#> set vpn tunnel tunnel=3 host_mode=enabled host_address=50.1.1.1
#> set vpn tunnel tunnel=3 host_mode_security=enabled
#> set vpn tunnel tunnel=3 remote_peer_address=57.42.65.21
#> set vpn tunnel tunnel=3 remote_tunnel_addr=192.168.20.0 remote_tunnel_
mask=255.255.255.0
#> set vpn tunnel tunnel=3 local_tunnel_addr=172.16.1.0 local_tunnel_
mask=255.255.255.0
#> set vpn tunnel tunnel=3 aggressive_mode=disabled
#> set vpn tunnel tunnel=3 natt_enable=enabled
#> set vpn tunnel tunnel=3 natt_ka_interval=20
#> set vpn tunnel tunnel=3 pfs=enabled
#> set vpn tunnel tunnel=3 dh_group_phase2=2
#> set vpn phase1 tunnel=3 proposal=1 state=disabled
#> set vpn phase1 tunnel=3 proposal=1 auth_method=rsa_sig
#> set vpn phase1 tunnel=3 proposal=1 authentication=sha1
#> set vpn phase1 tunnel=3 proposal=1 encryption=3des
#> set vpn phase1 tunnel=3 proposal=1 encryption_size=0
#> set vpn phase1 tunnel=3 proposal=1 sa_lifetime=28800
#> set vpn phase1 tunnel=3 proposal=1 sa_lifetime_data=50000
#> set vpn phase1 tunnel=3 proposal=1 diffie_hellman_group=2
#> set vpn phase1 tunnel=3 proposal=1 state=enabled
#> set vpn phase2 tunnel=3 proposal=1 state=disabled
#> set vpn phase2 tunnel=3 proposal=1 authentication=md5
#> set vpn phase2 tunnel=3 proposal=1 encryption=AES
#> set vpn phase2 tunnel=3 proposal=1 encryption_key_length=128
#> set vpn phase2 tunnel=3 proposal=1 sa_lifetime=28800
#> set vpn phase2 tunnel=3 proposal=1 sa_lifetime_data=50000
#> set vpn phase2 tunnel=3 proposal=1 state=enabled
#> show vpn tunnel tunnel=3 verbose=on
```

```
VPN Tunnel #3 Configuration :

   General Settings :

      name                : Tunnel 3
      mode                : isakmp
      autostart           : enabled
      host mode           : enabled
      host mode security  : enabled
      remote peer address : 57.42.65.21
      remote peer ID      : Certificate-DN
      interface           : mobile0
      local peer ID       : localpeer.digi1.com
```

```
Tunnel Settings :

   remote side         : ipv4subnet 192.168.20.0 - 255.255.255.0
   local side          : host address 50.1.1.1
       restricted to   : ipv4subnet 172.16.1.0 - 255.255.255.0


ISAKMP Settings:

    Client              : enabled
    Server              : enabled
    NAT Traversal       : enabled
    NAT-T KA Interval   : 20
    Aggressive mode     : disabled
    PFS                 : enabled
    Phase 1 DH Group    : set in each phase 1 proposal
    Phase 2 DH Group    : 2 (1024-bit)

ISAKMP Phase 1 Settings:

   index#  encryption/size  authentication
   ------  ---------------  --------------
   1       3des/0           sha1

Phase 2 Settings :

   index#  state     encryption  authentication
   ------  --------  ----------  --------------
   1       enabled   aes         md5
   2       disabled  des         md5
   3       disabled  des         md5
   4       disabled  des         md5
   5       disabled  des         md5
   6       disabled  des         md5
   7       disabled  des         md5
   8       disabled  des         md5
```

See also

- certmgmt
- revert: The **revert vpn** option reverts groups of VPN settings, or all VPN settings.
- set nat
- set network
- show vpn: Several **show vpn** command variants show current VPN settings in a Digi device.
- vpn: The **vpn** command is used to manage and display the status of VPN tunnels.
- The VPN settings in the web interface (**Network > Virtual Private Network (VPN) Settings**) and the online help for these settings.
- The *User Guide* for your Digi device, in the section titled "**Virtual Private Network (VPN) Settings**."

- Internet Engineering Task Force (IETF) document IETF RFC 3715, *IPsec-Network Address Translation (NAT) Compatibility Requirements* for information on NAT traversal.

set vrrp

Purpose

Configures Virtual Router Redundancy Protocol (VRRP) settings. VRRP allows several routers on a subnet to use the same virtual IP address. In this configuration setup, two or more physical routers are configured to represent the virtual router with only one doing the actual routing at any given time. If the current physical router fails, another physical router automatically replaces it. The advantage in using a virtual router redundancy protocol is that systems can be configured with a single default gateway, rather than running an active routing protocol.

Digi devices can support up to 8 virtual routers.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-vrrp=read** to display settings, and **set permissions s-vrrp=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure VRRP

```
set vrrp [index=1..8]
  [state={enabled|disabled}]
  [interface=eth0]
  [vrid=1..255]
  [priority={1..254}]
  [advertisement_interval=100..60000 (ms)]
  [preempt={enabled|disabled}]
  [ip_address=IP address of virtual router]
```

Display current VRRP settings

```
set vrrp
```

Options

index=1..8

The index number for a virtual router. Up to **8** virtual routers can be defined.

state={enabled|disabled}

Enables or disables the VRRP feature.

interface=eth0

The Ethernet interface on which VRRP advertisements should be sent and received.

vrid=1..255

The Virtual Router Identifier (VRID). All routers in the same VRID communicate with each other. The VRID can be any value between **1** and **255**. All routers that are to communicate must have the same VRID.

priority={1..254}

Priority determines which router is the master. The router with the highest priority is the master. The default priority is **100**.

advertisement_interval=100..60000 (ms)

The amount of time in milliseconds between VRRP master advertisements. All routers in the virtual routing group should be set to the same value. **3000** milliseconds (**3** seconds) is typically used.

preempt={enabled|disabled}

Controls whether a higher priority backup router preempts a lower priority master. Enter enabled to enable preemption; enter disabled to prohibit preemption. The default setting is enabled.

ip_address=*IP address of virtual router*

The IP Address of the virtual router. All routers in the same VRID should use the same virtual IP address. Clients should be configured to use this value as the default gateway.

Example

This command sets up a virtual router with an IP address of **192.168.1.101**, ID **10**, and priority **100**, advertising at **3 sec. intervals**:

```
#> set vrrp index=1 state=enabled vrid=10 priority=100 advertisement_
interval=3000 preempt=disabled ip_address=192.168.1.101
```

See also

- revert: The **revert vrrp** option reverts the settings configured by this command.
- show: The **show vrrp** command shows the current Virtual Router Redundancy Protocol settings in a Digi device.

set wimax

Purpose

Configures settings for the WiMAX radio in the Digi device. These settings control the current state of the radio, and its behavior when the Digi device is started.

Required permissions

For Digi products with two or more users, to use this command, permissions must be set to **set permissions s-wimax=read** to display settings, and **set permissions s-wimax=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax

Configure WiMAX radio

```
set wimax [state={on|off}]
  [auto_connect={on|off}]
  [nspid=id]
  [name=string]
  [eap_method={tls|ttls-chap|ttls-mschapv2|ttls-md5}]
  [username=string]
  [password=string]
  [realm=string]
  [disable_certs={on|off}]
```

Display current WiMAX radio settings

```
set wimax
```

Options

state={on|off}

Enables or disables the WiMAX radio.

on

Turn on the radio, scan for available networks, and be ready to connect.

off

Disables the WiMAX radio. If the radio is disabled, it will not transmit or receive over the air.

auto_connect={on|off}

Enables or disables automatic connection at startup to the subscription specified by the **nspid** or **name** option, and re-establish a connection if it is lost.

nspid=*id*

The identifier of the network service provider used in the automatic connection to the WiMAX network.

name=*string*

The name of the subscription or account with the network service provider used in the automatic connection to the WiMAX network.

eap_method={tls|ttls-chap|ttls-mschapv2|ttls-md5}]

The authentication method used for the automatic connection to the WiMAX network.

tls

Transport Layer Security (TLS). A client certificate and private key on the radio will be used to authenticate. This is the most common method used.

ttls-chap

Tunneled Transport Layer Security (TTLS) using the Challenge-Handshake Authentication Protocol (CHAP).

ttls_mschapv2

TTLS using Microsoft Challenge-Handshake Authentication Protocol Version 2 (MSCHAPV2).

ttls-md5

TTLS using Message Digest Version 5 (MD5).

username=*string*
password=*string*
realm=*string*

The username, password, and service realm name used for logging on to the WiMAX network.

If your service provider has given you account login information, specify the authentication method type using the **eap_method** option and enter the username, password, and realm values.

If you have a login of the form ***username@realm***, enter the user name and realm in separate fields, without the **@** sign.

When the authentication method is Transport Layer Security (TLS) (**eap_method=tls**) the **username**, **password**, and **realm** are not used.

disable_certs={on|off}

Disable server certificate verification.

Example

```
#> set wimax auto_connect=on nspid=000002 name=Clear eap_method=tls
```

See also

- display wimax
- revert: The **revert wimax** option reverts the settings configured by this command.
- show: The **show wimax** command displays current information and statistics about the WiMAX radio in a Digi device.
- wimax

- *Digi Quick Note: Digi Connect WAN 4G and ConnectPort Sprint/CLEAR 4G Configuration*, available on digi.com.

set wlan

Purpose

For a Digi device with Wi-Fi capability, configures wireless settings, or displays the status of wireless devices.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-wlan=read** to display settings, and **set permissions s-wlan=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Configuring wireless settings

Following is information on how configuration choices for wireless devices, such as the authentication method, affect other configuration choices, such as encryption types and other **set wlan** command options.

Authentication methods and available encryption types

The following table shows the authentication methods available for wireless devices, and the encryption types that apply to each method. The **X**s show the encryption types that can be used with each authentication method. At least one encryption type must be selected if a particular authentication method is selected.

Encryption Type:	Authentication Method:					
	Open	Shared Key	WEP authentication	WPA-PSK authentication	WPA	LEAP
Open	X	X				
WEP	X	X	X	X	X	X
TKIP				X	X	
CCMP				X	X	

Using "show wlan" to display authentication encryption methods

The **show wlan** command displays evaluation information about wireless LAN settings, including ineffective settings and a list of valid combinations. It displays whether encryption methods are specified and in use or not used by authentication methods, and whether configuration of certain options appears to be complete. For the results of **show wlan**, see the Examples section for show.

Authentication methods and associated data fields

The following table shows the authentication methods available for wireless devices, and the associated data fields, or command options that apply to each method. All data fields with that have an **X** in a particular authentication method's column are required, except for trusted certificates, which is optional.

Data Fields:	Authentication Method: Open	Shared Key	WEP authentication	WPA-PSK	WPA authentication	LEAP
WEP keys	X If WEP encryption is selected.	X				
Passphrase				X		
Authentication methods			X		X	
Username, password			X		X	X
Client certificate			X If TLS is selected.		X If TLS is selected.	
Trusted certificates			X		X	

Inner and outer protocols

The following table shows relationships between outer protocols and inner protocols specified on the **set wlan** command. Outer protocols are the types of Extensible Authentication Protocols (EAP) that are allowed to establish the initial connection with an authentication server or access point. The outer protocols are specified by the **outer_eap** option. Inner protocols are the types of protocols that are allowed to authenticate the device. These protocols are used within the encrypted connection established by PEAP or TTLs. The inner protocols are specified by the **inner_eap** option.

Inner Protocols:	Outer Protocols: PEAP	TLS	TTLS
GTC	X		
MD5	X		X
MSCHAPv2	X		X
OTP	X		X
TLS	X		X
CHAP			X
MSCHAP			X
MSCHAPv2			X
PAP			X

Syntax

Configure wireless settings

```
set wlan
  [protmode={bss|ibss_create|ibss_join|any}]
  [channel={0|1-14}]
  [ssid=string]
  [authentication={[open],[sharedkey],[wep_auth],[wpa_psk],
  [wpa_auth],[leap],[any]}
  [encryption={[open],[wep],[tkip],[ccmp],[any]}]
  [outer_eap={[peap],[tls],[ttls],[any]}]
  [inner_eap={[gtc],[tls],[md5],[mschapv2],[otp],[chap],[mschap],
  [ttls_mschapv2],[pap],[any]}]
  [options={[diversity],[short_preamble],[verify_cert]}]
  [username=string]
  [password=string]
  [psk=string]
  [psk_hex=hex string]
  [wepmode={64bit|128bit}]
  [wepindex=1-4]
  [wepkeyN=hex string]]
  [country=”string”]
  [maxtxrate={1|2|5.5|6|9|11|12|18|24|36|48|54}] (Mbps)
  [txpower={6|8|10|12|14|16}] (dBm)
```

Display wireless settings

```
set wlan
```

Or:

```
show wlan
```

Options

Command options authentication, encryption, outer_eap, inner_eap, and options can have multiple values

The **set wlan** options **authentication, encryption, outer_eap, inner_eap, and options** can have multiple values. More than one value may be specified for each option to indicate the set of allowed values. The actual value used is determined by the capabilities of the wireless network.

protmode={bss|ibss_create|ibss_join|any}

Used to change the operation mode in which the device works.

bss

Indicates that the device should join an access point.

ibss_create

Indicates the device will attempt to first join an Independent Basic Service Set (IBSS) or ad hoc wireless network, and create one if it is unable to find one.

ibss_join

Indicates the device should attempt to join an IBSS or ad hoc wireless network.

any

Enables all operation modes.

Typically, the operation mode is **bss**. The default is **bss**.

channel={0|1-14}

Sets the frequency channel that the wireless LAN radio uses. A value of **0** indicates that the device scans all frequencies until it finds one with an available access point or wireless network it can join. The default value is **10**.

ssid=*string*

Used to specify the identifier of the wireless network that the device should be joined to. The default is an empty string, which indicates that the first wireless network that the device finds will be joined to.

authentication=
{[open],[sharekey],[wep_auth],[wpa_psk],[wpa_auth],[leap],[any]}

The types of authentication that are allowed to establish a connection with the access point.

open

Use the IEEE 802.11 open system authentication to establish a connection with the access point.

sharedkey

Use the IEEE 802.11 shared key authentication to establish a connection with the access point. At least one WEP key must be specified to use shared key authentication.

wep_auth

IEEE 802.1x authentication (EAP) is used to establish a connection with an authentication server or access point. Wired Equivalent Privacy (WEP) keys are dynamically generated to encrypt data over the wireless link.

wpa_psk

Use the The Wi-Fi Protected Access (WPA) protocol with a pre-shared key (PSK) that you specify to establish a connection with the access point and encrypt the wireless link.

wpa_auth

Use the The WPA protocol and IEEE 802.1x authentication (EAP) to establish a connection with an authentication server or access point. Encryption keys are dynamically generated to encrypt data over the wireless link.

leap

Use the Lightweight Extensible Authentication Protocol (LEAP) to establish a connection with an authentication server or access point. Wired Equivalent Privacy (WEP) keys are dynamically generated to encrypt the wireless link. A username and password must be specified to use leap.

any

Sets all authentication types.

encryption={[open],[wep],[tkip],[ccmp],[any]}

The types of encryption that are allowed to encrypt data transferred over the wireless link.

open

Use No encryption over the wireless link. Can be used with **open** and **sharedkey** authentication.

wep

Use Wired Equivalent Privacy (WEP) encryption over the wireless link. Can be used with **open**, **sharedkey**, **wep_auth**, **wpa_psk**, **wpa_auth**, and **leap** authentication.

tkip

Use Temporal Key Integrity Protocol (TKIP) encryption over the wireless link. This can be used with **wpa_psk** and **wpa_auth** authentication.

ccmp

Use CCMP (AES) encryption over the wireless link. Can be used with **wpa_psk** and **wpa_auth** authentication.

any

Sets all encryption types.

outer_eap={[peap],[tls],[ttls],[any]}

The types of Extensible Authentication Protocols (EAP) that are allowed to establish the initial connection with an authentication server or access point. These are used with **wep_auth** and **wpa_auth** authentication.

peap

Protected Extensible Authentication Protocol (PEAP). A username and password must be specified to use peap.

tls

Transport Layer Security (TLS). A client certificate and private key must be installed on the device to use tls.

ttls

Tunneled Transport Layer Security (TTLS). A username and password must be specified to use ttls.

any

Sets all outer and inner Extensible Authentication Protocols (EAP).

inner_eap={[gtc],[tls],[md5],[mschapv2],[otp],[chap],[mschap],[ttls_mschapv2],[pap,[any]]}

The types of protocols that are allowed to authenticate the device. Use these within the encrypted connection established by PEAP or TTLS.

The following are Extensible Access Protocols (EAP) that can be used with PEAP or TTLS:

gtc

Generic token card.

tls

Transport Layer Security (TLS). A client certificate and private key must be installed on the device to use tls.

md5

Message Digest Algorithm (MD5).

mschapv2

Microsoft Challenge response Protocol version 2.

otp

One Time Password.

The following are non-EAP protocols that can be used with TTLS:

chap

Challenge response Protocol.

mschap

Microsoft Challenge response Protocol.

ttls_mschapv2

Microsoft Challenge response Protocol version 2.

pap

Password Authentication Protocol.

any

Sets all inner Extensible Authentication Protocols.

options={[diversity],[short_preamble],[verify_cert]}

diversity

Enable reception on multiple antennas on devices with this capability.

short_preamble

Enable transmission of wireless frames using short preambles, if allowed by the access point.

verify_cert

Verify that certificates received from an authentication server or access point are signed by a trusted certificate authority (CA). Standard CAs are built in, and additional trusted certificates may be added.

username=*string*

Used when the **security** option is set to **wep_auth**, **wpa_auth**, or **leap**. This option specifies the user name to be used during authentication.

password=*string*

Used when the **security** option is set to **wep_auth**, **wpa_auth**, or **leap**. This option specifies the password to be used during authentication.

psk=*string*

Used when the **security** option is set to **wpa_psk**. This option specifies a string that is converted into a pre-shared key (PSK) that is used for encryption.

psk_hex=*hex string*

Used when the **authentication** option is set to **wpa_auth**. **psk** and **psk_hex** are alternate ways of setting the PSK. This option specifies the hexadecimal value of the pre-shared key (PSK) used for encryption. The key consists of **64** hexadecimal digit characters.

wepmode={64bit|128bit}

Specifies the key size used when WEP encryption is enabled. The default is **64bit**.

wepindex=1-4

Specifies which of the 4 possible keys will be used. The default is **1**.

wepkeyN=*hex string*

A hexadecimal string that serves as the key if WEP encryption is enabled. The key consists of 26, 10, or 0 (zero) hexadecimal digit characters. If **wepmode=64bit**, the wepkey is **10** digits. If **wepmode=128bit**, the wepkey is **26** digits. A wepkey value of **0** length clears any value that was previously set.

country=*string*

The country where the device will be used. By selecting a country, the channel settings are restricted to the legal set for that country. Allowed country names are:

United States, Australia, Austria, Belgium, Canada, China, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Japan, Liechtenstein, Luxembourg, Netherlands, Norway, Poland, Portugal, Singapore, Spain, Sweden, Switzerland, United Kingdom

Note Country names that include spaces should be enclosed in quotation marks; for example, "**United States**".

maxtxrate={1|2|5.5|6|9|11|12|18|24|36|48|54} (Mbps)

The maximum transmission rate that the device uses, in megabits per second. The complete range of transmission rates is available on all devices except the ConnectPort X2 - XBee to Wi-Fi model. For that model, the allowed transmission rates are **{1|2|5.5|11}**.

txpower={6|8|10|12|14|16} (dBm)

The wireless transmit power, in decibels relative to one milliwatt (dBm).

Example

```
#> set wlan wepkey1=ab12cd34ef567ab12cd34ef567 wepindex=1
```

```
#> set wlan wepmode=128bit
```

```
#> set wlan ssid=”access point 1"
```

See also

- revert: The **revert wlan** option reverts the settings configured by this command.
- show: The **show wlan** command displays an evaluation of saved wireless settings, including ineffective settings and a list of valid combinations. It displays whether encryption methods are specified and in use or not used by authentication methods, and whether setup of certain options appear to be complete.

set xbee

Purpose

For Digi devices with an embedded XBee RF module, the **set xbee** command performs several tasks:

- Changes or displays the gateway's handling of XBee functionality, including enabling/disabling XBee network communications for the gateway and setting parameters for the XBee RF module. See Syntax and options.
- Changes or displays settings for an XBee RF module on an XBee network node. See Syntax and options.
- Configures XBee firmware update settings on the gateway. See Configure XBee firmware update settings on the gateway. The actual firmware updates are performed by the **xbee** command; see xbee.
- Sends AT commands, either to the local XBee RF module on the gateway or on network nodes. AT commands are used to set or view parameters for XBee RF modules. See Syntax and options.
- Saves device configuration settings for nodes to a backup file, that can be used to restore the device configuration settings if the need ever arises. See Syntax and options.
- For ConnectPort X2 gateways, to communicate directly with the XBee RF module on the gateway, known as Direct Access. See Direct Access communication with the XBee RF module on ConnectPort X2 gateways.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-mesh=read** to display settings, and **set permissions s-mesh=rw** to display and configure settings. See set permissions for details on setting user permissions for commands.

Syntax and options

Display available settings for XBee RF module on the gateway

```
help set xbee
```

Configure XBee settings on the gateway

These settings control the behavior of the gateway rather than the XBee RF module.

```
set xbee [state=(on|off)]
```

Where:

state={off|on}

Enables or disables XBee network communications for the gateway.

Display XBee settings for a node

```
set xbee [address={node id|network address|extended address}]
```

Where:

address={*node id*|*network address*|*extended address*}

The address of the node, specified by its node ID, 16-bit network address, or extended address.

To display the node list, enter a **display xbee** command, or go to this page in the web interface: **Configuration > XBee Network.** If the address is not specified, the device settings for the local XBee module on the gateway are displayed.

Configure XBee firmware update settings on the gateway

You can update XBee RF moduleswith new firmware over the XBee network. XBee firmware updates are available through Digi Technical Support and are loaded onto the gateway through the web interface.

There are two kinds of XBee firmware updates:

- A *gateway firmware update*, which is an update of the XBee firmware in the Digi device serving as a gateway for an XBee network. You can perform gateway firmware updates on any type of XBee RF module.
- An *OTA (Over the Air) firmware update,* which is an update of the XBee firmware in the XBee network nodes. As XBee networks can involve a large number of nodes, Digi provides a way to schedule automatic XBee firmware updates and manage firmware files. You can perform OTA firmware updates on XBee ZB RF modules.

The **xbee** command controls the actual firmware updates See xbee.

The syntax for configuring XBee firmware update settings on the gateway is:

```
set xbee [fw_update={on|off}]
  [fw_automatic={on|off}]
  [fw_stop_on_error={on|off}]
```

Where:

fw_update={on|off}

Enables/disables the XBee firmware update process.

Note **fw_update** causes extra traffic on the network that might interfere with applications. It is off by default and must be turned on to enable over-the-air (OTA) firmware updates.

fw_automatic={on|off}]

Enables/disables automatic XBee firmware updates.

fw_stop_on_error={on|off}]

Stop on firmware update error.

Uploading and managing XBee firmware files

The command line interface does not have a way to upload and manage XBee firmware files. Uploading and managing firmware files must be done from the gateway's web interface, as shown in the process below.

Updating XBee ZB firmware on gateway and nodes

The following process applies to OTA firmware updates for XBee ZB module types only. For other module types, or if you want to update the gateway XBee module, you have to use the gateway firmware update page in the web interface. That page uploads a single file and does the update in a single step.

To update XBee firmware on XBee ZB modules, follow these steps:

1. Load the required XBee firmware files from Digi's Support page onto a PC. Firmware files for ZB nodes have an **.ebl** extension.
2. Upload the XBee firmware files from the PC to the gateway's web interface. Go to **Configuration > XBee Network > Firmware Update Setup**

   Multiple files can be uploaded, each containing a different firmware type needed by nodes on the network.
3. Identify the nodes to be updated, then schedule and monitor updates of individual nodes on the **Firmware Update Status** page of the web interface, or use the **set xbee** commands described later in this command description.

   Each scheduled update is performed in the background, one node at a time. While a remote node is being updated, it is inaccessible from the XBee network. While the XBee module in the gateway is being updated, the XBee network is inaccessible from the gateway.

Start an XBee firmware update immediately after firmware files are uploaded

The following is the syntax for starting an XBee firmware update immediately after firmware files are uploaded:

```
set xbee fw_automatic
```

Where:

fw_automatic

Sets firmware updates to start as soon as the firmware files are uploaded, or as nodes are discovered, without having to schedule them to be updated. An appropriate firmware version is automatically selected from the uploaded files. To restart after an error, schedule another update using the **xbee fw_update** option or the **Configuration > XBee Network > OTA Firmware Update Status** page of the web interface.

Settings preserved during firmware updates

If you enable the gateway, most XBee module settings are preserved during the firmware update. Some settings, such as encryption keys, may not be preserved and must be entered again.

XBee Firmware requirements and file naming conventions

XBee firmware updates involve several requirements and follow naming conventions identifying the file by hardware, network, and node type. For details, see the section titled **Firmware updates for XBee modules** in the User's Guide for your ConnectPort X gateway.

View the status oHf XBee firmware updates

To view the status of XBee firmware updates, use the **xbee fw_update** command. See page 501.

Configure XBee RF module settings on a node

```
set xbee [address={node id|network address|extended address} [device_settings]
```

Where:

address={*node id*|*network address*|*extended address*}

The address of the node, specified by its node ID, 16-bit network address, or extended address. If the address is not specified, settings are changed on the local XBee RF module in the gateway.

device_settings

The firmware configuration settings available on the XBee RF module. These settings can vary by XBee RF module type and firmware level. To display available settings:

set xbee ? displays available settings for the XBee RF module in the gateway in the form of help text.

set xbee displays the actual settings for the XBee RF module in the gateway.

set xbee address={*node id*|*extended address*} to display settings for the XBee RF module on another node.

For descriptions of the settings, see the *Product Manual* for the XBee RF module, available from the Support site on digi.com.

Settings are saved to XBee non-volatile RAM (NVRAM) automatically.

Send an AT command to a node

The syntax for sending an AT command to an XBee node is:

```
set xbee [address={node id|network address|extended address}] CC[=parameter]
```

Where:

address={*node id*|*extended address*}

The address of the node where the device sends the AT command, specified by its node ID, 16-bit network address, or extended address. If the address is not specified, the AT command is sent to the local XBee module on the gateway.

***CC*[[=]*parameter*]**

The AT command to be sent to the node, where **CC** is the AT command, specified as two upper-case characters. For example, SM is the command for setting sleep mode.

AT commands vary among node types and the XBee RF module protocol running on the node. To display the available AT commands for both the gateway and remote nodes, enter **help set xbee**. See the *Product Manual* on digi.com for the XBee RF module for detailed AT command descriptions.

[[=]*parameter*]

The associated parameters for the AT command.

Parameters included with the AT command can be:

- A decimal value, for example, **4**
- A hexadecimal value, specified as **0x<hex>**, for example, **0x4**
- A string value, for example, **"string"**

If no parameter is specified, and the AT command is a settings command, the **set xbee** command displays the current value for the setting.

Save settings to non-volatile RAM

To save device settings to non-volatile RAM (NVRAM), send the node the AT command **WR**. The **WR** command saves settings to the XBee node specified with **address=*xxx***. The **WR** command is only needed when using AT commands.

Options for backing up and restoring XBee RF module settings

You can save the XBee RF module configuration settings for nodes to a backup file and used them to restore the configuration settings if the need ever arises.

Back up XBee RF module settings for a node to a file

The backup operation saves the node's XBee RF module configuration settings to a file on a TFTP server. The resulting backup file is a **.pro** file that is compatible with the XCTU configuration tool. This means that backup files can be saved or loaded from the XBee RF module using XCTU as well as the gateway's command line or web interfaces.

The syntax for the backup operation is:

```
set xbee to=server[:filename]
```

Where:

server

The host name or IP address of the TFTP server.

filename

The name of the backup file. The default filename is address.pro where address is the address of the node, for example: **0013a200403294bb.pro**.

Restore XBee RF module settings for a node from a file

The restore operation sets the node's XBee RF module configuration settings to those in the specified .**pro** file from a TFTP server.

CAUTION! A restore operation can cause the device to reset its network information, reset, and rejoin a network, meaning it may no longer be accessible from this gateway.

The syntax for the restore operation is

```
set xbee from=server[:filename]
```

Where:

server

The host name or IP address of the TFTP server.

filename

The name of the backup file. The default filename is **address.pro** where address is the address of the node, for example: **0013a200403294bb.pro**.

Direct Access communication with the XBee RF module on ConnectPort X2 gateways

On ConnectPort X2 gateways, it is possible to directly communicate with the XBee RF module through its serial port, rather than through the XBee driver software on the gateway. This type of direct access is targeted for applications that require communicating directly with the XBee RF module in the ConnectPort X2 gateway via RealPort, Modbus, or UDP/TCP sockets; to do so, the XBee driver must be disabled. Once disabled, the port used for the XBee driver is open for other services to directly access the XBee RF module, including RealPort, TCP/UDP Sockets, Modbus, and use an open Python serial port.

1. Disable the XBee driver on the gateway using the **set xbee** command:

   ```
   set sbee state=off
   ```

2. Depending on the service used to access the port, enter additional configuration commands or implement additional programming to use the port for direct access to the XBee RF module.

   To use RealPort to access the port:

   a. Enter a **set service** command with no options to display all the available network services on the gateway:

   ```
   set service
   ```

   b. In the command output, note the index number assigned to the RealPort service. For example:

   ```
   #> set service
   Service Configuration :
   index state ipport keepalive nodelay           service
   8     on    771       off       na                      RealPort Service
   ```

   c. Enter another **set service** command, specifying the index number assigned to the RealPort service on the **range** option:

```
set service range=range state=on
```

For example:

```
#> set service range=8 state=on
```

Once the RealPort service is enabled, the ConnectPort X2 gateway acts as the RealPort server. See the *RealPort User's Guide* on digi.com for more details on using the RealPort service.

To use TCP/UDP sockets to access the port:

a. Use the **set profile** command to configure the port with the settings in the **tcp sockets** or **udp_sockets** port profile:

```
set profile tcp_sockets
```

Or:

```
set profile udp_sockets
```

b. The default baud rate for the XBee RF module is **115200**. Use the **set serial** command to set the baud rate to **115200** and set the flow control to use **hardware** handshaking.

```
set serial baudrate=115200 flowcontrol=hardware
```

To use an open Python serial port to access the port:

Using open Python serial ports to access the port requires additional programming. See the *Digi Python Programming Guide* and the description of the **termios** module.

Examples

Disable the XBee gateway

```
#> set xbee state=off
```

Enable OTA firmware updates

```
#> set xbee fw_update=on
```

Set node ID string on gateway

```
#> set xbee node_id="gateway"
```

Set node ID string on a remote node

```
#> set xbee address=00:13:a2:00:40:32:94:bb! node_id="adapter"
```

Set node ID string on a remote node using AT commands

```
#> set xbee address=00:13:a2:00:40:32:94:bb! NI="adapter" WR
```

Backup remote node's settings

```
#> set xbee address=adapter to=my_server:config_file.pro
```

Restore remote node's settings

```
#> set xbee address=adapter from=my_server:config_file.pro
```

See also

- The *Product Manual* for the XBee RF module in the Digi device located on digi.com. This manual contains configuration settings, descriptions of status information, operation details, and descriptions of AT commands that can be sent to XBee RF modules.
- display Xbee shows the list of nodes in the same XBee network and within the range of the XBee RF module in the Digi device, or status information on a particular XBee node.
- info xbee: This command displays statistics from the ConnectPort X gateway's perspective of what is happening on the ZigBee network.
- revert: The **revert xbee** command reverts the settings configured by this command.
- show: The **show xbee** command shows the current XBee settings for the XBee RF module in a Digi gateway.
- xbee
- These pages of the web interface for the Digi device have the same configuration settings as the "set xbee" command:
  Configuration > XBee Network > Firmware Update Setup
  Configuration > XBee Network > Firmware Update Status
- The "TCP Sockets" profile can be used to directly access the XBee RF module on a gateway. See set profile and the Direct Access communication example in this command description.
- The RealPort service can be used to directly access the XBee RF module on a gateway. See set service and the Direct Access communication example in this command description.
- *RealPort User's Guide* on digi.com for more details on using the RealPort service.

  Digi Python Programming Guide, available on the Digi website.

show

Purpose

Displays the current settings in a device, including current configuration settings, boot code loaded in the device, and the effects of commands issued to the device.

Required permissions

For Digi products with two or more users, for this command to display current device settings, the various **set** commands that configure the settings displayed by a **show** command must be set to either **read** or **r-self**, depending on the available permissions for the commands. See set permissions for details on setting user permissions for commands.

Syntax

```
show options [options]
```

Options

***options* [*options*]**

Specifies which settings in the device to show, for example, **show time**. The following table shows the options can be specified.

options

Are any additional options, which depend on the settings group.

Throughout these descriptions, the following descriptions apply to the **port** and **range** options:

port=*range*

Identifies a particular serial port. Optional on a single-port device.

range=*range*

A configuration table entry or range of entries.

show *option* value and additional options	Displays settings configured by
accesscontrol	**set accesscontrol**
alarm	**set alarm**
arp	The ARP table. This option is not associated with a **set** command.
autoconnect	**set autoconnect**

show *option* value and additional options	Displays settings configured by
buffer	**set buffer**
camera	**set camera**
clocksource [**range=1-5]** **Note** The ranking value of the unconfigurable real-time clock source is listed for reference purposes.	**set clocksource**
dcloud_msgservice	**set dcloud_msgservice**
ddns	**set ddns**
devicesecurity	**set devicesecurity**
dhcpserver	**set dhcpserver**
dialserv [ports=1..1]	**set dialserv**
dirp [range=1-20]	**set dirp**
dnsproxy	**set dnsproxy**
ekahau	**set ekahau**
ethernet There are several command variants: ■ **show ethernet all** displays a summary of all Ethernet settings. ■ **show ethernet interface=*comma-separated list of interface names*** displays interface detail for the specified interfaces. ■ **show ethernet interface=*** displays interface detail for all interfaces. ■ **show ethernet** with no options is equivalent to **show ethernet all**. **if** can be used as an abbreviation for **interface**.	**set ethernet**
failover [interface=*interface name*] Valid Interface names are: **mobile0**,**eth0** **if** can be used as an abbreviation for **interface**. If **interface** is not specified, settings are shown for all interfaces.	**set failover**
forwarding [interface=*interface name*] Valid interface names are: **mobile0**, **eth0** **if** can be used as an abbreviation for **interface**. If **interface** is not specified, settings are shown for all interfaces.	**set forwarding**
g**eofence [index=1-16]**	**set geofence**

show *option* value and additional options	Displays settings configured by
gpio	**set gpio**
group[id=1-32] (group ID)[name=*group name*]	**set group**
host	**set host**
hostlist	**set hostlist**
ia master	**set ia** command: settings for an IA master; see Configure network-based masters—"set ia master".
ia serial	**set ia** command: settings for IA serial; see Configure serial-port connected devices—"set ia serial".
ia table	**set ia** command: settings for IA destination tables and route entries; seeConfigure destination tables and route entries—"set ia table".
idle	**set idle**
login	**set login**
menu	**set menu**
mgmtconnection	**set mgmtconnection**
mgmtglobal	**set mgmtglobal**
mgmtnetwork	**set mgmtnetwork**
mobile [index=1-2]	**set mobile**
mobileppp [index=1-2]	**set mobileppp**
module	**set module**
nat [instance=1-8] **[showformat={list | tabular}]** **instance** specifies which NAT instances should be displayed. If an instance is not specified, all instances will be shown. **showformat** specifies how NAT instances should be displayed: as a list or in table form. Default **showformat** is **tabular**.	**set nat**

show *option* value and additional options	**Displays settings configured by**
network There are several command variants: **show network all** displays a summary of all network settings. **show network interface=*comma-separated list of interface names*** displays interface detail for the specified interfaces. **show network interface=*** displays interface detail for all interfaces. **if** can be used as an abbreviation for **interface**. **show network** with no options is equivalent to **show network all**.	**set network**
orbcomm	**set orbcomm**
passthrough	**set passthrough**
permissions [type={user|group}][id=1-32] [name={*user name*|*group name*}] **set permissions**	**set pmodem**
pmodem	
position	**set position**
pppinbound	**set ppp**
ppp [port=1-5]	**set ppp**
profile	**set profile**
putty	**set putty**
rciserial	**set rciserial**
realport	**set realport**
realportusb	**set realportusbusb**
route	The IP routing table. This command is not associated with a **set** command.
rtstoggle	**set rtstoggle**
scancloak	**set scancloak**
serial	**set serial**
service	**set service**
sharing	**set sharing**
slideshow	**set slideshow**
smscell See show smscell.	**set smscell**
snmp	**set snmp**

show *option* value and additional options	Displays settings configured by
socket_tunnel	**set socket_tunnel**
surelink [index=1-2]	**set surelink**
switches [port=range]	**set switches**
system	**set system**
tcpserial	**set tcpserial**
term	**set term**
time	**set time**
udpserial [range=1-64]	**set udpserial**
user [id=1-32] [name=user name]	**set user**
versions	This command shows firmware version information. It is not associated with a **set** command.
video	**set video**
vncclient	**set vncclient**
vpn [options...] See show vpn.	**set vpn**
vrrp [index=1-8]	**set vrrp**
wimax	**set wimax**
wlan	set wlan This option displays an evaluation of saved wireless settings, including ineffective settings and a list of valid combinations. It displays whether encryption methods are specified and in use or not used by authentication methods, and whether setup of certain options appear to be complete (see Examples).
xbee [address={*id*|address}]	**set xbee**

Examples

Display network configuration settings

```
#> show network

Network configuration:

MAC Address          : 00:40:9D:24:8B:B3
```

```
                      Currently in use by
                      the network stack      Stored configuration
                      --------------------   -------------------
ipaddress           : 10.8.16.8              192.168.4.25
submask             : 255.255.0.0            255.255.0.0
gateway             : 10.8.1.1               0.0.0.0
static              : off                    off
dhcp                : supplied IP address    on
autoip              : on                     on
keepalive idle      : 7200                   7200
probe count         : 9                      9
probe interval      : 75                     75
garbage byte        : on                     on
verride dhcp        : off                    off
dns1                : 10.10.8.62             0.0.0.0
dns2                : 10.10.8.64             0.0.0.0
rto_min             : 1000                   1000
rto_max             : 10                     10
arp_ttl             : 15                     15
garp                : 3600                   3600
```

Display current alarm settings

```
#> show alarm
```

Display settings for a particular user

```
#> show user range=3
```

Display wireless settings

In addition to showing the current wireless settings, **show wlan** displays evaluation notes and warning messages about the effect and interaction of wireless settings. As the example shows, warning messages note encryption methods that have been defined but not used by any authentication methods, and notes identify whether configuration of certain features appears to be complete.

```
#> show wlan

Wireless LAN Configuration:

                   Active settings
                  (Stored settings)
                  --------------------
country         :  United States
protocol mode   :  any
channel         :  scan
ssid            :
maxtxrate(Mbps) :  11
authentication  :  open
encryption      :  open,wep,tkip,ccmp
eap outer       :  peap,tls,ttls
eap inner       :  gtc,md5,mschapv2,otp,chap,mschap,ttls_mschapv2,pap
options         :
txpower(dBm)    :  14dbm
wepmode         :  64bit
```

```
wepindex        :  1
username        :

Evaluation of your saved wireless settings:

Warning: TKIP encryption specified but unused by any of the specified
   Authentication methods
Warning: CCMP encryption specified but unused by any of the specified
   Authentication methods
Note: Settings for Protocol Modes IBSS Create or IBSS Join appear
   to be complete
Note: Settings for Open Authentication appear to be complete
```

See also

- revert
- The **set** commands (**set user**, **set network**, **set serial**, and so on.). Entering a **set** command without any options displays the same information as that displayed by the **show** command.
- set wlan for the encryption and authentication methods and protocol modes referenced in the **show wlan** output.

show smscell

Purpose

Displays current settings for the Short Message Service (SMS) capabilities of the mobile module of the Digi device.

Required permissions

For Digi products with two or more users, for this command to display settings, permissions for the **set smscell** command must be set to **set permissions s-sms-cellular=read** or **set permissions s-sms-cellular=rw**. See set permissions for details on setting user permissions for commands.

Syntax

```
show smscell [group]
```

Options

group

One or more SMS settings groups, configured by several versions of the **set smscell** command. If no group is specified, **all** is the default.

global

Show global SMS settings that configure operation of SMS for the Digi device.

scl

Show Sender Control List SMS settings. The Sender Control List (SCL) permits users to select the addresses (or phone numbers) from which SMS messages will be accepted.

python

Show settings for processing Python commands sent via SMS.

command

Show settings controlling execution of built-in SMS commands. Several built-in commands are supported for execution via SMS messages sent to your Digi device.

all

Show all SMS settings.

The default is **all**.

Examples

```
#> show smscell

Global options:

  state           : off
  ackrcvdcmds     : off
  nakpswdfail     : off
  exteventlog     : off
  password        : (none)
  defreceiver     : logonly
  cmdidchar       : #

SCL options:

  state           : off
  nakrejectedcmds : off
  SCL entry 1:
    enabled       : off
    match         : right
    address       :
  SCL entry 2:
    enabled       : off
    match         : right
    address       :
  SCL entry 3:
    enabled       : off
    match         : right
    address       :
  SCL entry 4:
    enabled       : off
    match         : right
    address       :
  SCL entry 5:
    enabled       : off
    match         : right
    address       :
  SCL entry 6:
    enabled       : off
    match         : right
    address       :
  SCL entry 7:
    enabled       : off
    match         : right
    address       :
  SCL entry 8:
    enabled       : off
    match         : right
    address       :
  SCL entry 9:
    enabled       : off
    match         : right
    address       :
  SCL entry 10:
    enabled       : off
    match         : right
```

```
    address       :
  SCL entry 11:
    enabled       : off
    match         : right
    address       :
  SCL entry 12:
    enabled       : off
    match         : right
    address       :
  SCL entry 13:
    enabled       : off
    match         : right
    address       :
  SCL entry 14:
    enabled       : off
    match         : right
    address       :
  SCL entry 15:
    enabled       : off
    match         : right
    address       :
  SCL entry 16:
    enabled       : off
    match         : right
    address       :
```

```
Python options:

  state           : on
  password        : (none)
  readqueuemsgmax : 100
  readholdsecmax  : 600
```

```
Command options:

  Command "help":
    state         : on
    password      : (none)
  Command "cli":
    state         : on
    password      : (none)
  Command "dcloud":
    state         : on
    password      : (none)
  Command "ping":
    state         : on
    password      : (none)
```

See also

- display smscell
- set smscell
- smscell

show vpn

Purpose

Displays VPN configuration settings. Keywords allow for displaying all VPN settings or specified groups of settings.

Required permissions

For Digi products with two or more users, for this command to display current device settings, permissions for the **set vpn** command must be set to **read** or **read/write** by the **set permissions s-vpn=read** or **set permissions s-vpn=rw** commands. See set permissions for details on setting user permissions for commands.

Syntax

Show all VPN configuration settings

```
show vpn all
```

Show VPN global settings

```
show vpn global
```

Show VPN tunnel settings

```
show vpn tunnel [tunnel=1-5]
  [name=tunnel name]
  [verbose={on|off)]
```

Show VPN IKE/ISAKMP SA Phase 1 options for tunnels

```
show vpn phase1 [tunnel=1-5)
  [name=tunnel name]
  [verbose={on|off}]
```

Show VPN IKE/ISAKMP SA Phase 2 options for tunnels

```
show vpn phase2 [tunnel=1-5]
  [name=tunnel name]
  [proposal=1-8]
  [verbose={on|off}]
```

Show the network interface used to communicate with the remote VPN device

```
show vpn interface
```

Options

tunnel=1-5

Selects the VPN tunnel by number.

name=*tunnel name*

Selects the VPN tunnel by name.

verbose={on|off}

If set to **on**, a detailed list of settings will be displayed. If set to **off,** a short summary of the tunnel settings is displayed.

proposal=1-8

The index number assigned to the security proposal.

Examples

Display VPN tunnel configuration summary

This example shows how to display a summary of VPN tunnel configuration:

```
#> show vpn tunnel tunnel=1 verbose=off

 VPN Tunnel Configuration :

#  name              remote endpoint  remote tunnel       local tunnel
-  ----------------  ---------------  ------------------  ----------------
1  Tunnel 1          75.75.75.75      192.168.1.0/24      172.16.1.0/24
```

Display detailed VPN tunnel configuration settings

This example shows how to display the detailed list of configuration settings for a tunnel:

```
#> show vpn tunnel tunnel=1 verbose=on

 VPN Tunnel #1 Configuration :

    General Settings :

       name                : Tunnel 1
       mode                : isakmp
       autostart           : disabled
       host mode           : disabled
       remote peer address : 75.75.75.75
       remote peer ID      :
       interface           : mobile0
       local peer ID       : walter@digi.com
```

```
    Tunnel Settings :

       remote side         : ipv4subnet 192.168.1.0 - 255.255.255.0
       local side          : ipv4subnet 172.16.1.0 - 255.255.255.0
```

```
    ISAKMP Settings:
```

```
        Client              : enabled
        Server              : enabled
        NAT Traversal       : enabled
        NAT-T KA Interval   : 20
        Aggressive mode     : enabled
        PFS                 : enabled
        Phase 1 DH Group    : set in each phase 1 proposal
        Phase 2 DH Group    : 2 (1024-bit)
```

```
    ISAKMP Phase 1 Settings:

       index#  encryption/size  authentication
       ------  ---------------  --------------
       1       3des/0           md5
```

```
Phase 2 Settings :

       index#  state     encryption  authentication
       ------  --------  ----------  --------------
       1       enabled   3des        md5
       2       disabled  des         md5
       3       disabled  des         md5
       4       disabled  des         md5
       5       disabled  des         md5
       6       disabled  des         md5
       7       disabled  des         md5
       8       disabled  des         md5
```

See also

- display vpn
- set vpn
- vpn

smscell

Purpose

Sends a message to a destination via Short Message Service (SMS).

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-sms-cellular=rw** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
smscell [sendmsg=to,text]
```

Options

sendmsg=*to*,*text*

Sends the specified message to the specified destination.

to

The message destination.

text

The message text.

Examples

```
#> smscell sendmsg=622639,b
```

See also

- display smscell
- set dcloud_msgservice
- set smscell

status

Purpose

Displays the current list of sessions. The **status** command displays the status of outgoing connections (connections made by **connect**, **rlogin**, or **telnet** commands). In contrast, the **display** command displays real-time information about a device, while the **info** command displays statistical information about a device over time. Typically, the **status** command is used to determine which sessions to close.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions status=read** or **set permissions status=rw** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
status [range] [session number]
```

Options

range

The range of sessions to view.

session number

An index number identifying the session number to view.

Examples

```
#> status

  Connection: 3          From: 10.8.109.8

    Connection not associated with any sessions.
```

See also

- connect
- close for information on ending a connection.
- “display” commands.
- “info” commands.
- rlogin
- telnet
- who

telnet

Purpose

Used to make an outgoing Telnet connection, also known as a session.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions telnet=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

telnet [options] [{ip address|hostname}] [tcp port]

Options

options

The Telnet options for the command, which may be as follows:

binary={on|off}

Turns on or off Telnet binary mode.

crmod={on|off}

Turns on or off the replacement of the carriage-return character sequence (**\r**) with the new-line character sequence (**\n**) on incoming network data.

{*ip address*|*hostname*}

The IP address of the host to which you want make a Telnet connection.

tcp port

The TCP port assigned the Telnet application on the remote system. The default is **23**, the port typically used for Telnet.

Examples

Establish a Telnet session using an IP Address

In this example, the telnet command establishes a Telnet session using an IP address. The default TCP port (**23**) is used.

```
#> telnet 192.192.150.28
```

Establish a Telnet session to a device server port from the LAN

In this example, a user on the LAN initiates a Telnet connection to port **4** on a Digi device.

```
#> telnet 192.192.150.28 2004
```

See also

- rlogin
- connect
- close
- status

vpn

Purpose

Manages and displays the status of a Virtual Private Network (VPN) tunnel.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions vpn=execute** to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
vpn [{status|connect|disconnect}]
  [index=1-2)]
  [name=tunnel name]
```

Options

{status|connect|disconnect}

The action performed by the command. You must attempt a data connection before bringing up the VPN tunnel.

status

Display the status of the VPN tunnel.

connect

Enable the VPN tunnel.

disconnect

Disconnect the VPN tunnel.

index=1-2

Identifies the VPN tunnel.

name=*tunnel name*

The name of the tunnel.

Examples

Display VPN tunnel status

```
#> vpn status

 VPN Tunnel #1 Status :

    name           : Tunnel 1
```

```
    mode            : isakmp

    status          : down
    remote address: : 65.214.122.53
    mobile address: : not connected
```

```
VPN Tunnel #2 Status :

    name            : Tunnel 2
    mode            : disabled
```

Enable a VPN tunnel

This command enables use of the tunnel at index **1**.

```
#> vpn connect index=1
```

See also

These commands display VPN-related connection and status information:

- display ikesa
- display ikespd
- display ipsecspd
- display sadb
- display spd
- display vpn
- revert. The **revert vpn** options revert groups of VPN settings, or all VPN settings.
- set vpn. The **set vpn** command configures VPN settings.
- The VPN settings in the web interface (**Network > Virtual Private Network (VPN) Settings**) and the online help for these settings.
- The **Connections Management** page in the web interface (from the **Home** page, click the **Connections** link) is the web equivalent of this command.
- The *User's Guide* for your Digi device, in the section titled **Virtual Private Network (VPN) Settings**.

watchport

Purpose

Queries Watchport sensors attached to the Digi device and displays their current sensor data.

This command functions for Watchport sensors for Drop-in Networking only. These sensors have a model number ending with **-20**. Watchport sensors with a model number ending with **-01** sensors do not support this command.

Syntax

```
watchport query
```

Options

query

Queries Watchport sensors attached to the Digi device and displays their current sensor data.

Examples

```
#> watchport query
   Humidity: 2.717 %
   Temperature: 23.6 deg C  74.5 deg F
```

who

Purpose

Displays active connections to and from the device. For Digi devices that have Python programs loaded and running, the **who** command can be used to view which Python threads are running. Python threads are listed for informational purposes. They cannot be killed with either the **kill** command or via the **Management > Connections** page in the web interface.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions who=execute**" to use this command. See set permissions for details on setting user permissions for commands.

Syntax

```
who
```

Options

None at this time.

Examples

Display a list of all current connections

```
#> who
```

Display Python threads

The Python program **EmbeddedKitService.py** is running on a ConnectPort X device. Entering **who** displays this output:

```
#> who

ID  From                 To                   Protocol              Sessions
--  -------------------  -------------------  --------------------  --------
1   10.8.16.27           local shell          telnet
2                                             Python: EmbeddedKitService.py

3                                             Python thread
4                                             Python thread
5                                             Python thread
6                                             Python thread
7                                             Python thread
8                                             Python thread
9                                             Python thread
10  10.8.16.27           local shell          telnet
```

Next, the **EmbeddedKitService.py** program is stopped on the ConnectPort X device. Entering **who** displays this output:

```
#> who

ID   From                  To                    Protocol              Sessions
--   --------------------  --------------------  --------------------  --------
1    10.8.16.27            local shell           telnet
#> who

ID   From                  To                    Protocol              Sessions
--   --------------------  --------------------  --------------------  --------
1    10.8.16.27            local shell           telnet
```

See also

- kill to kill a connection.
- python to execute Python programs.
- set python to configure Python programs.
- who to view which Python threads are running.
- The *Digi Python Programming Guide* to learn more about the Python programming language as implemented in Digi products, and writing Python programs.
- The Python Support Forum on digi.com.

wimax

Purpose

For Digi devices equipped with a WiMAX radio, **wimax** performs various WiMAX network actions, including:

- Connect to a WiMAX network.
- Disconnect from a WiMAX network.
- Set debug options for testing and diagnostics.
- Display detailed WiMAX radio information.
- Perform a wide-area scan, an over-the-air discovery of networks that are available for connections. The radio automatically scans for networks in the subscription list. A wide-area scan looks for all networks that might be available. The scan takes a few minutes. The results of either automatic or wide-area scan are shown under the network list by issuing a **display wimax** command.

Required permissions

For Digi products with two or more users, permissions must be set to **set permissions s-wimax=rw** to execute this command. See set permissions for details on setting user permissions for commands.

Syntax

Connect to a WiMAX network

First enter **display wimax t**o get the ***subindex*** from the subscription displayed, and the ***napid*** from the network list. Then, enter:

```
wimax connect [[subindex [napid]]
```

Disconnect from a WiMAX network

```
wimax disconnect
```

Set debug options

```
wimax debug {dm|eap}
```

Display detailed WiMAX radio information

```
wimax info
```

Reset the WiMAX radio

```
wimax reset
```

Perform a wide-area scan

```
wimax scan
```

Options

subindex

Subscription index. A subscription is the service and user account with a network service provider (NSP). This value can be obtained from the output of the **display wimax** command, under the **Subscription List Index** heading. The subindex defaults to using the **nspid** and **name** set by the **set wimax** command.

napid

Network access provider ID. This value provides the actual network connection and is usually the same as the network service provider (NSP). The default behavior WiMAX and NAP IDs is to try all networks found by the scan. Using the **napid** option allows for choosing specific network.

debug {dm|eap}

Turns on various debugging functions. Debugging is normally used for testing and diagnostics, not by a user.

dm

Diagnostic monitor protocol. This option is used with external tools to control and monitor the radio.

eap

Extensible Authentication Protocol. This option enables EAP debugging. It writes debug information to two files: **dec.log** and **pcap.pcap**.

Examples

```
#> wimax info

Connection Information:
  Radio Status: Connected
  Connection Duration: 1 day + 05:50:35
  Disconnect Reason: N/A
  Subscription Name: Clear
  Network Type: Home
  NAP-ID: 000002
  RSSI: -76 dBm
  CINR: 9 dB
  Signal Quality: 2 of 5 bars
```

```
Network Information:
  IP Address: 184.78.91.72
  Gateway: 184.78.0.1
  Primary DNS: 66.233.235.12
  Secondary DNS: 75.94.255.12
```

```
Radio Information:
  Manufacturer: GCT Semiconductor, Inc.
```

```
  Model: Quanta WM553
  MAC Address: 00:17:C4:9C:53:F9
  SW Version: 1.10.1.2
  FW Version: 2.0.0.4
  HW Version: 0.0.7.0
```

```
Subscription List:
  INDEX  OPERATOR      NAME          NSP-ID  ACTIVATED
      1  Clear         Clear         000002  yes
      2  Clear         Sprint 4G     000002  yes
      3  Clear         Sprint PCS    000002  yes
```

```
Network List:
  NAME         TYPE     NAP-ID  RSSI  CINR
  Clear        Home     000002   -88     6
```

```
Neighbor List:
  BSID           FREQUENCY  PREAMBLE  RSSI  CINR
  000002:264FA2    2657000        40   -89     3
  000002:264FA3    2667000        72  -123   -10
  000002:2628FB    2551500        79   -89     6
  000002:264BE1    2647000        25  -123   -10
  000002:264BE2    2657000        57   -93    -1
  000002:264EEB    2667000        66  -123   -10
  000002:264F23    2667000        69  -123   -10
  000002:264F22    2647000        37   -85   -10
```

```
RF Information:
  BSID: 000002:2628FB
  UL PermBase: 8
  DL PermBase: 8
  Current preamble index: 8
  Previous preamble index: 0
  HO count: 0
  HO fail count: 0
  Resync count: 0
  HO signal latency: 89
  Combined CINR: 9 dB
  CINR: 6 dB
  CINR2: 6 dB
  Combined RSSI: -76 dBm
  RSSI: -79 dBm
  RSSI2: -79 dBm
  PER: 0.000161 [1/6211]
  Power control mode: 1
  TX power: 18 dBm
  TX power maximum: 23 dBm
  TX power headroom: 9 dBm
  UL burst data FEC scheme: QPSK (CTC) 3/4
  DL burst data FEC scheme: QPSK (CTC) 1/2
  UL burst data UIUC: 02
  DL burst data DIUC: 00
  Frequency: 2647000 KHz
  Power mode: Idle
```

```
Statistics:
  Session TX Bytes:  2046
  Session TX Frames: 13
```

```
Session RX Bytes:  80729
Session RX Frames: 1088
Total TX Bytes:    2046
Total TX Frames:   13
Total RX Bytes:    80729
Total RX Frames:   1088
Connections: 1
Connection Failed: 0
Authentication Failed: 0
User Requested: 0
Network Disconnect: 0
Radio Reset: 0
```

See also

- display wimax
- set wimax
- show: The **show wimax** command displays information and statistics about the WiMAX radio.

xbee

Purpose

Executes an XBee utility or displays the status of actions performed by the XBee utilities. The actions include displaying information about the XBee network setup, sending loopback data, displaying the status of XBee firmware, and scheduling, canceling, and viewing status of XBee firmware updates.

Some variants of the **xbee** command are available on certain XBee networks only.

- The following XBee utilities are only available on XBee ZB and Xbee SE networks:
  - **Display child table**: Displays a list of end devices that are joined to a router node.
  - **Display neighbor table**: Displays a list of nodes one hop away from the specified node.
  - **Display source route**: Displays the source route received from a node when many-to-one routing is being used.
  - **Display route**: Displays the route between two nodes when ad-hoc on-demand routing is being used. The default source node is the gateway.
- The following XBee utilities are available on all XBee networks except XBee 802.15.4 and XBee SE:
  - **Display identify messages**: Identify messages are used to determine the physical location of remote nodes. This XBee utility displays identify messages on the command line as they are received from remote nodes. A remote node may send this message when a user presses its commissioning button, or by some other means.
  - **Send identify message**: Sends an identify message to a remote node. This action may cause the remote node to blink its LED, or give some other indication it received the message.
  - **Send loopback data**: Tests communication with a remote node.This action sends a message that the node repeats back to the gateway.
- The following XBee utilities are only available on XBee ZB networks:
  - **Schedule an XBee firmware update:** Schedules XBee firmware updates for the XBee RF module on the gateway and on XBee network nodes.
  - **Cancel an XBee firmware update**: Cancels a previously scheduled XBee firmware update.
  - **View the status of XBee firmware updates**: Displays status of scheduled firmware updates for all nodes, both the gateway and remote nodes.

Syntax and options

Display child table

```
xbee child_table node
```

Where:

node

A node serving as a router in an XBee network, specified by its node ID, 16-bit network address, or extended address.

Display neighbor table

```
xbee neighbor_table node
```

Where:

node

A node serving as a router in an XBee network, specified by its node ID, 16-bit network address, or extended address.

Display source route

```
xbee source_route node
```

Where:

node

An XBee network node, specified by its node ID, 16-bit network address, or extended address.

Display route

```
xbee route destination [source]
```

Where:

source

The source node to use as the start of the displayed route. The route to display starts at the source node and ends at the destination node. The default source node is the gateway.

Display or turn off identify messages

```
xbee identify {on|off}
```

Send identify message

```
xbee identify node [seconds]
```

Where:

node

The node to which an identify message is sent, specified by its node ID, 16-bit network address, or extended address. Identify messages are used to determine the physical location of remote nodes

seconds

The duration of the identification; for example, the length of time an LED displays a blink pattern.

Send loopback data

```
xbee ping node [count] [size] [interval_ms]
```

Where:

count

The number of messages to send. The default is 4 messages.

size

The number of bytes in each message. The default is 16 bytes.

interval_ms

The interval between messages, in milliseconds. The default is 1000 milliseconds.

Schedule an XBee firmware update

The syntax for scheduling an XBee firmware update is:

```
xbee fw_update target [updater] [file]
```

Where:

target

The node to update, specified either by its node ID or extended address.

updater

An optional nearby node to control the firmware update. The default is to choose a node automatically.

file

The file containing the XBee firmware update, ending in an **.ebl** extension. The default is to automatically select a file from the files that have been uploaded to the gateway, if possible.

Cancel an XBee firmware update

```
xbee fw_update target cancel
```

target

Node for the firmware update is canceled, specified either by its node ID or extended address.

cancel

Indicates that the firmware update for the specified node should be canceled.

View the status of XBee firmware updates

```
xbee fw_update
```

This command shows a table of scheduled firmware updates for all nodes (both the gateway and remote nodes) and the status. Status fields that display are listed in the following table:

Field	Description
Node ID	The user-assigned identifier of the node.
Extended Address	The unique 64-bit MAC address of the node.
HW	The hardware type and version of the node. **XBP** indicates that the node is an XBee-PRO module. **S2B** indicates an XBee-PRO S2B node. **S2C** indicates an XBee S2C node. **S2CP** indicates an XBee PRO S2C node.
FW	The current firmware version of the node.
Status	The firmware update status of the node. It may be one of these values: **Unknown**: Current firmware version has not yet been read from the node, or cannot be read from the node. **Up to date**: The node is running the latest firmware version available on the Digi device server. **Available**: A newer version of firmware is available on the gateway. Schedule an update using the "xbee fw_update" command or the **Configuration > XBee Network > OTA Firmware Update Status** page of the web interface. **Scheduled**: A firmware update is scheduled to be performed on this node. **Updating**: A firmware update is now being performed on this node. Updated: A successful firmware update has been performed on this node. **Complete**: The node has rejoined the network after a successful firmware update. **Cancelled**: A firmware update for this node has been cancelled by a user. **Error**: A firmware update on this node has failed. Schedule an update using the "xbee fw_update" command or the **Configuration > XBee Network > OTA Firmware Update** Status page of the web interface
File	The firmware file used to update the node.

Examples

Ping a remote node:

```
#> xbee ping adapter
```

Update firmware on a node:

```
#> xbee fw_update adapter firmware_file.ebl
```

Display firmware update status:

```
#> xbee fw_update
```

See also

- display Xbee
- info xbee
- info zigbee_sockets
- set xbee
- For more information about the XBee utilities executed by this command, see the XBee *Product Manual* for the XBee modules in your gateway and nodes on digi.com, and the ZigBee specification.

Modem Emulation Commands

This chapter describes the commands that can be issued when Digi devices are configured in modem emulation mode.

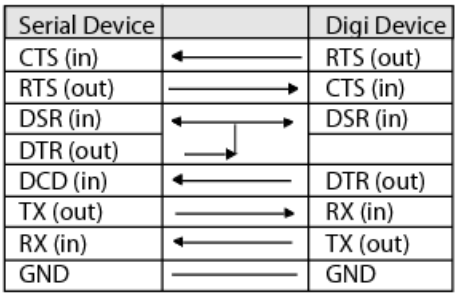
What Is Modem Emulation?

Modem emulation enables a system administrator to configure a networked Digi device to act as a modem. The Digi device emulates modem responses to a serial device and seamlessly sends and receives data over an Ethernet network instead of a PSTN (Public Switched Telephone Network). The advantage for a user is the ability to retain legacy software applications without modification and use a less expensive Ethernet network in place of public telephone lines.

As an aid in configuring modem emulation, the Digi Device Setup Wizard and the default web interface have a serial port profile for modem emulation.

Modem Emulation Cable Signals

Use the following signal assignments to make a cable connecting the Digi device to a serial device.



DSR and **DTR** on the serial device side are connected to the DSR signal of the Digi device.

Modes of Operation

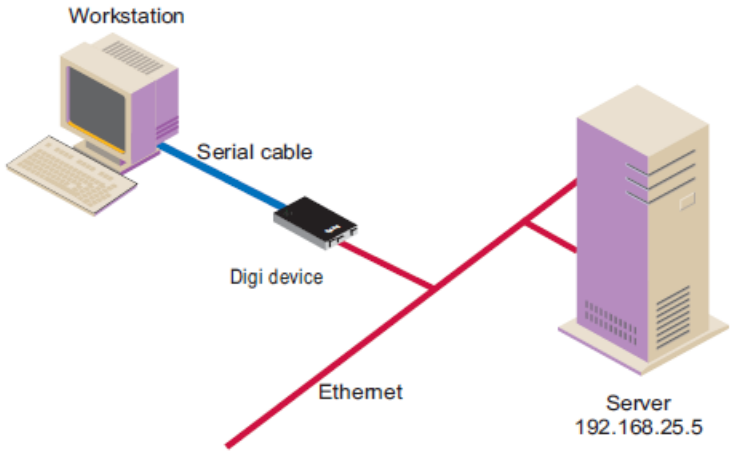
There are two modes of operation in modem emulation:

- Command mode: Issuing AT commands to a Digi device.
- Data mode: After a network connection is established, the device switches to data mode.

Common User Scenarios for Modem Emulation

The Digi device in modem emulation mode allows for the easy replacement of modems in almost any environment where there is a LAN or WAN.

User Scenario - Diagram A

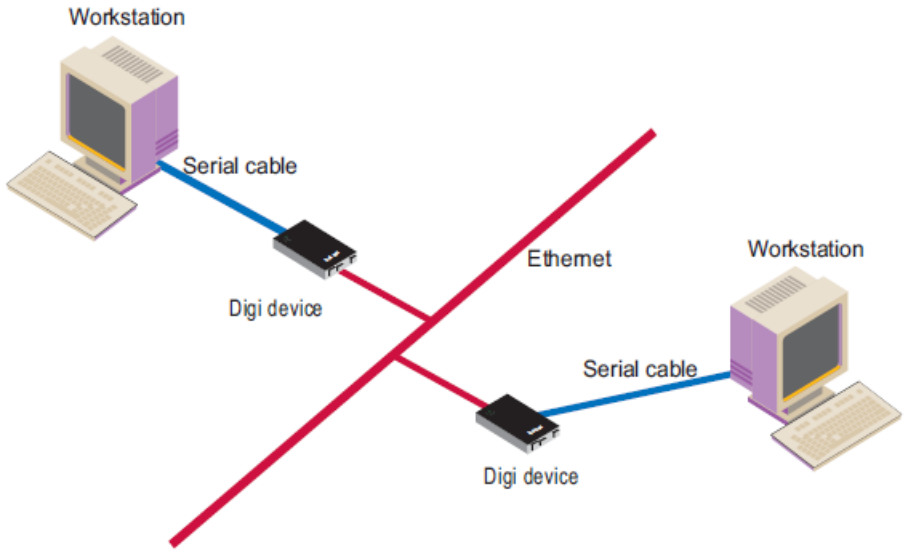


In Diagram A, the Digi device replaces a modem connected to a workstation running an application. The Digi device allows you to use software applications without modification by responding to all the AT commands configured in the workstation application. The Digi device connects to the IP Address of the server when you issue the following command:

```
ATDT ipaddress:port (ATDT 192.168.25.5:50001)
```

Once the remote device establishes the TCP connection, a **CONNECT** message is sent to the serial port and only then does the Digi device switch from AT command mode to data mode. Using the modem escape sequence or dropping **DTR** on either side terminates the connection. A **DISCONNECT** message is sent to the application if the remote side closes the TCP connection.

User Scenario - Diagram B



In Diagram B, two Digi devices replaces modems on both sides of the connection. The initiation of the connection occurs with either of the Digi devices. If both ends are Digi devices, the TCP listening port number is **50001** for port **1**. An example of the connection command is:

```
ATDT 192.168.25.30:50001.
```

Upon establishing a successful TCP connection, a **CONNECT** message is sent to the serial port, and only then does the Digi device switch from AT command mode to data mode. After the **CONNECT** is received, the transmission of data begins. Using the modem escape sequence or dropping **DTR** on either side terminates the connection.

Connection Scenarios for Modem Emulation

Modem emulation can involve the following types of connection scenarios:

Outgoing Modem Emulation Connection

In an outgoing modem emulation connection, a serial device sends an **ATD*x.x.x.x:y*** command, which triggers the Digi device to establish a connection to destination **IP=*x.x.x.x***, **port=*y***.

Incoming Modem Emulation Connection

In an incoming modem emulation connection, a device on the network connects to port **50001** (**50000**+**1** = 1st serial port). This incoming connection triggers the Digi device to generate a **RING** on the serial port. The device attached to the serial port answers the **RING** and the connection is established.

Modem Emulation Pooling

Modem emulation pooling is a combination of Incoming Modem Emulation Connection and a hunt group. A device on the network connects to port **50000**. The Digi device checks if a serial port configured for modem emulation is available. If so, it connects to the port, otherwise returns an error.

Modem Emulation Bridge

A modem emulation bridge is combination of Outgoing and Incoming Modem Emulation Connections, in which both serial devices require to talk to a modem. The first serial device connects to the second device using **ATD*x.x.x.x:y***, the second device gets a **RING** and accepts the incoming connection.

About the Commands in this Chapter

This chapter describes the Digi-specific modem emulation commands that have been implemented for Digi devices. It is divided into several sections:

- The AT command set. These are commands to perform actions in a modem-emulation connection.
- Modem S-Register definitions.
- A description of the result codes for the commands.

Accepted But Ignored AT Commands

Any other commands not described in this chapter but in the standard AT command set are accepted but ignored and therefore have no effect. Such commands are pertinent to actual modems, but not to modem emulation.

Modem Emulation AT Command Set

The following commands can be issued to perform actions in a modem-emulation configuration scenario.

AT Command	Function	Result Code
n+++n	When in data mode, this command causes the modem to switch to command mode. The value of **n** corresponds to the required delay before and after the escape sequence is entered. The delay can be changed by modifying S-register 12. The escape character can be changed by modifying S-register 2.	
A/	Repeats the last command string.	
AT?	Prints the value of the last-accessed S-register.	
ATA	Answer command: Answers an incoming TCP connection and switches to data mode.	
ATD (ipaddress): (ipport)	Used to connect to a remote network device. This command directs the Digi device to go on-line, dial according to the IP address entered as follows, and attempt to establish a TCP connection. Dial Modifiers. The valid dial string parameters are described below. Punctuation characters may be used for clarity with parentheses, hyphen, and spaces being ignored. ■ **0-9**: DTMF digits 0 through 9. ■ **.** (period): Dot notation used for IP addresses. IP addresses are written as four numbers separated by periods, where the first number is between 1 and 255, and the other three numbers are between 0 and 255. Enter the IP address in the format **xxx.xxx.xxx.xxx** ■ **:** (colon): Colon notation used for the TCP port. ■ **L**: Redial the last number. The modem will reconnect to the last IP address accessed. The **L** must immediately follow the **D**, and any following characters are ignored. ■ **P**: This command is accepted but not acted on. ■ **T**: This command is accepted but not acted on. ■ **R**: This command is accepted but not acted on. ■ **,** (comma): This command is accepted but not acted on.	

AT Command	Function	Result Code
ATEn	Command echo. The Digi device enables or disables the echo of characters to the DTE according to the parameter supplied. The parameter value, if valid, is written to S14 bit **1**. ■ **E0**: Disables command echo. ■ **E1**: Enables command echo.	OK n=0 or 1 ERROR Otherwise
ATH	Disconnect (Hang up) command. **H0**, **H1**: Hangs up the TCP connection if a connection is active.	OK n=0 or 1 ERROR Otherwise
ATIn	Identification command. ■ **I0**, **I1**: Reports product name. ■ **I3**: Reports product name, firmware revision. ■ **I4**: Reports product configuration. ■ **I6**: Reports network connection information.	OK n=0 or 9 ERROR Otherwise
ATO	Return to on-line data mode. If the modem is in the on-line command mode, the modem enters the on-line data mode. If the modem is in the off-line command mode (no connection), **ERROR** is reported. ■ **O0**, **O1**: If there is an active connection, switches the modem to data mode.	OK n = 0 or 1 and a connection exists. ERROR Otherwise or if not connected.
ATQn	Quiet results codes control command. The command enables or disables the sending of the result codes to the DTE according to the parameter supplied. The parameter value, if valid, is written to S14 bit 2. **Q0**: Enables result code to the DTE (Default). **Q1**: Disables result code to the DTE. **Q2**: Disables **CONNECT** result codes. **Q3**: Disables **CONNECT** result codes on incoming connections.	OK n=0 or 1 ERROR Otherwise
ATSn	Read/Write to the specified S-Register. ■ **n** Establishes S-register **n** as the last register accessed. ■ **n=v** Sets S-Register **n** to the value **v**. ■ **n?** Reports the value of S-Register **n**. See S-Register Definitions for definitions of S-Registers.	OK n=0 or 1 ERROR Otherwise

AT Command	Function	Result Code
ATVn	The verbose setting for result codes. This command selects the sending of short-form or long-form codes to the DTE. The parameter, if valid, is written to S14 bit 3. ■ **V0**: Result codes are issued in numeric or short form. Line feeds are not issued before a short-form result. ■ **V1**: Result codes are issued in text or long form. This is the default.	OK n=0 or 1 ERROR Otherwise
ATZ	Load configuration. Reloads the S-register configuration from flash memory. See S-Register Definitions for definitions of S-Registers.	OK n=0 or 1 ERROR Otherwise
AT&Cn	DCD option. The Digi device controls the **DCD** output in accordance with the parameter supplied. The parameter value, if vali,d is written to S21 bit 5. ■ **&C0**: DCD remains ON at all times. ■ **&C1**: DCD follows the state of the connection.	OK n=0 or 1 ERROR Otherwise
AT&Dn	DTR option. This command interprets the ON to OFF transition of the **DTR** signal from the DTE in accordance with the parameter supplied. The parameter value, if valid, is written to S21 bits 3 and 4. Also see S25. ■ **&D0**: DTR drop is ignored (assumed ON). ■ **&D1**: DTR drop is interpreted by the modem as if the asynchronous escape sequence had been entered. The modem returns to command mode without disconnecting. ■ **&D2**: DTR drop causes the modem to hang up. (Default.) ■ **&D3**: DTR drop causes the modem to do a soft reset, as if the **ATZ** command was executed.	OK n=0 to 3 ERROR Otherwise
AT&F	Restore factory configuration. The device reloads the factory default S-register configuration from flash memory. The factory defaults are identified for each command and in the S-Register descriptions. A configuration consists of a subset of S-Registers.	OK n=0 or 1 ERROR Otherwise
AT&V	Displays current values and settings. ■ **AT&V0**- **AT&V5**: Displays S-Register/command values for the current and stored configuration. ■ **AT&V6**: Displays current network settings.	OK n=0 to 5 ERROR Otherwise

AT Command	Function	Result Code
AT&Wn	Store configuration. Stores the specified S-registers in flash memory.	OK n=0 or 1 ERROR Otherwise

S-Register Definitions

Following are definitions of the S-registers that can be set using modem emulation commands.

Register	Function	Range	Units	Default
S0	Rings to Auto-Answer. Sets the number of rings required before the Digi device automatically answers a call. Setting this register to Zero disables auto-answer mode.	0-255	Rings	0
S1	Ring Counter. Specifies the current number of rings. **S1** is incremented each time the modem detects a ring signal on the telephone line. **S1** is cleared when the existing connection is established or dropped.	0-255	Rings	0
S2	Escape Character. **S2** holds the value of the ASCII character used as the escape character. The default value corresponds to an ASCII '**+**'. A value over 127 disables the escape process. That is, no escape character will be recognized.	0-255	ASCII	43
S3	Carriage Return Character. Sets the value of the carriage return character used when displaying commands or results.	0-127	ASCII	13
S4	Line Feed Character. Sets the character recognized as a line feed when displaying commands or results. If verbose result codes are used, the Line Feed control character is output after the Carriage Return control character.	0-127	ASCII	10
S5	Backspace Character. Sets the character recognized as a backspace, used to erase the last character typed on the command line.	0-32	ASCII	8
S12	Escape Prompt Delay. The amount of time required before and after an escape sequence (**+++**) is entered in order for the modem to transition from data mode to command mode.	0-255	0.02 second, 20 ms	50 1 second

Register	Function	Range	Units	Default
S14	General Options Status. Indicates the status of command options. ■ Default: **138 (8Ah) (10001010b)** ■ Bit 0: Ignored. ■ Bit 1: Command echo (En): 0 = Disabled (E0). 1 = Enabled (E1). (Default.) ■ Bits 2 and 4: Quiet mode (An): 0 = Display result codes (Q0). (Default.) 1 = Do not display result codes (Q1). 2 = Disables "**CONNECT**" result codes (Q2). 3 = Disables "**CONNECT**" result codes on incoming connections (Q3). ■ Bit 3: Result codes (Vn): 0 = Display numeric result codes (V0). 1 = Display verbose result codes (V1). (Default.) ■ Bits 5-7: Ignored.			138 (8Ah)

Register	Function	Range	Units	Default
S21	General Options Status. Indicates the status of command options. ■ Default: **52 (34h) (00110100b)** ■ Bits 0 - 2: Ignored. ■ Bits 3-4: The DTE's DTR behavior (**&Dn**): 0 = DTR drop is ignored (**&D0**). 1 = DTR drop causes a transition from data to command mode without hanging up an existing connection (**&D1**). 2 = DTR drop hangs up the existing connection (**&D2**) (Default.) 3 = DTR drop causes the modem to do a soft reset if the ATZ command was executed (**&D3)**. ■ Bit 5: Modem's DTR behavior: 0 = The modem's DTR remains on at all times (**&C0**). 1 = The modem's DTR follows the state of the TCP connection (**&C1**). (Default.) ■ Bits 6-7: Ignored.	-	-	52 (34h)
S25	Delay to DTR Off. The amount of time that the modem delays before taking the action specified by the **AT&Dn** command.	0-255	s or 0.01 s	5

Result Codes

Modem emulation commands return the following return codes.

Short	Long Form		Short	Long Form		Short	Long Form
0	OK		13	CONNECT 7200		84	CONNECT 33600
1	CONNECT		14	CONNECT 12000		91	CONNECT 31200
2	RING		15	CONNECT 14400		165	CONNECT 32000
3	NO CARRIER		16	CONNECT 19200		166	CONNECT 34000
4	ERROR		17	CONNECT 38400		167	CONNECT 36000
5	CONNECT 1200		18	CONNECT 57600		168	CONNECT 38000
6	NO DIALTONE		19	CONNECT 115200		169	CONNECT 40000
7	BUSY		20	CONNECT 230400		170	CONNECT 42000
8	NO ANSWER		59	CONNECT 16800		171	CONNECT 44000

Short	Long Form		Short	Long Form		Short	Long Form
9	CONNECT 0600		61	CONNECT 21600		172	CONNECT 46000
10	CONNECT 2400		62	CONNECT 24000		173	CONNECT 48000
11	CONNECT 4800		63	CONNECT 26400		174	CONNECT 50000
12	CONNECT 9600		64	CONNECT 28800			

B

C

D

E

F

G

H

K

N

O

P

Q

R

S

T

U

V

W

X