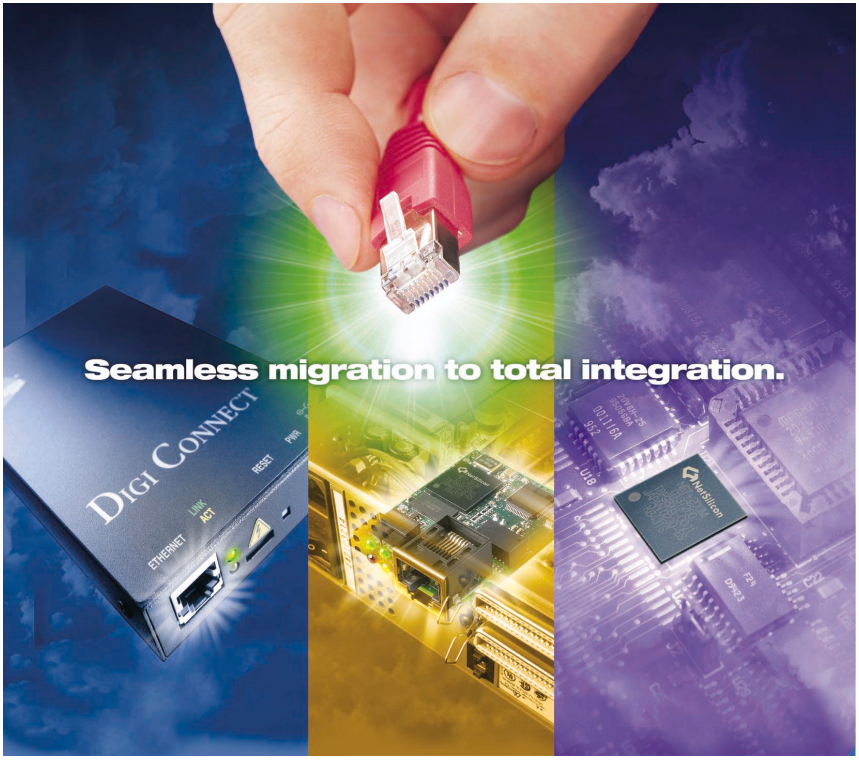
Digi®

Connectware™

NetSilicon
A Digi International Company

Remote Command Interface (RCI) Specification



Making DEVICE NETWORKING easy™

Table of Contents

1 Scope

This document describes the Remote Command Interface.

2 Terminology

The terminology used in this document follows XML definitions.

- An XML element is the *example_element* in this example: **<example_element>**
- An XML attribute is the example_attribute="value" in this example: **<example_element example_attribute="example_value">**, where *example_attribute* is the name of the attribute, and *example_value* is the value of the attribute. Note the double-quote characters are required.

3 Overview

RCI is a method for remote clients to control, configure, and gather statistics from Digi Connect devices. RCI is a stateless, request/response protocol. RCI uses XML and HTTP (or HTTPS) to exchange data between clients and Digi devices.

4 The RCI Protocol

The interface is split into three layers: Transport, Command, and Data.

4.1 Transport

The transport layer is a mechanism specified to handle communication between a client and a Digi device. The transport will specify the initialization process, the sending and responding mechanism, the closing mechanism, any error recovery mechanism needed, and security.

4.1.1 RCI over HTTP

The primary transport is HTTP, through the embedded web server. The Web server will provide the initialization, receiving and sending, and security.

RCI requests are sent to the device using an URI of UE/rci. For example, if the Digi Device's IP address is 192.168.1.1, then RCI requests are sent to http://192.168.1.1/UE/rci

RCI requests are sent as an HTTP POST with the XML request of the form specified in this document. Note, due to space limitations on the device, **the largest request that can be processed is 32KB**. If a request is larger than this, it must be split into multiple RCI requests. RCI replies from the device are not subject to this limit.

Security is handled in the usual HTTP mechanism. The username and password must be passed to the device in the header of each HTTP request.

HTTPS is also supported.

See the samples shipped with devices for examples of RCI over HTTP RCI.

Standard HTTP errors will be returned for HTTP related problems. Common HTTP errors that should be handled by clients:

- 413 - Buffer too large. Usually caused by sending a request larger than 32KB in size.

4.1.2 RCI over Serial

RCI requests can be sent over the serial port. This is useful in scenarios where a master processor is connected to the Digi Device through a serial port. This allows the master processor to configure the Digi Device as part of its configuration process, so that a separate, manual configuration step for the Digi Device is eliminated.

The Digi device must be placed in Configuration mode. Once in configuration mode, the Digi Device will accept RCI requests and return replies.

The method to use to place the device into configuration mode was not defined at the time this document was updated. Additional information will be provided in the release notes for the device firmware when this feature is available.

4.2 RCI Request/Reply

An RCI XML document is identified by the XML elements rci_request and rci_reply.

An RCI request specifies the XML element "rci_request" optionally with a version number. The version should match the version of RCI the client expects. The current RCI version is 1.1 If a version is not specified, the RCI version of the device is used to form the reply. Not specifying a version can cause problems when communicating with devices at different RCI versions, if the client code is not written in a version independent way. Therefore, **it is highly recommended to always supply the version of RCI in requests**, unless the client code has been designed to be version independent.

Example of a request element:
<rci_request version="1.1">

The device will respond to requests with the element "rci_reply" along with the version number as an attribute.

Example reply: <rci_reply version="1.1">

4.2.1 <rci_reply> Errors

Errors that occur at the request level will result in an error element as a sub-element of the <rci_reply>. Errors and warnings are explained below

<rci_reply> errors:

Error ID	Description
1	Request not valid XML
2	Request not recognized
3	Unknown command

4.3 Command

The command section of the protocol indicates the action requested (or action peformed in replies).

Commands are specified as sub-elements to <rci_request> and <rci_reply>.

This example requests all configuration settings:

```
<rci_request version="1.1"> <!–Identifies the protocol and whether this is a request or a response -->
   <query_setting/>        <!– request config of device -->
</rci_request>
```

This example requests the configuration information for just boot settings and serial settings.

```
<rci_request version="1.1">
   <query_setting>
      <boot/>
      <serial/>
   </query_config>
</rci_request>
```

Supported commands:

COMMNAD	REQUEST DESCRIPTION	RESPONSE DESC
query_setting	Request for device settings. May contain setting group elements to subset query (only setting group subset supported. Subseting below this level not supported).	Returns requested config settings. Requests specifying no settings groups (eg. <query_setting/>) return all settings.
set_setting	Set settings specified in setting element. Settings data required.	Empty setting groups in reply indicate success. Errors returned as specified below.
query_state	Request current device state such as statistics and status. Sub-element may be supplied to subset results.	Returns requested state. Requests specifying no groups (eg. <query_state/> return all state.
set_factory_default	Sets device settings to factory defaults. Same semantics as set_setting.	Same semantics as set_setting.
reboot	Reboots device immediately.	

4.4 Data

Data elements are elements used to exchange information about settings and state.

Data elements are typically of the form:

```
<data_group>
   <field>value</field>
</data_group>
```


Where “data_group” is a grouping of related fields, such as “serial” for serial settings. “field” are names of individual settings, such as “baud”. “value” is the value associated with the field.

Data is expressed in XML. Data is of two types:
setting: used for settings and control. Setting data may be read/write, read-only, or write-only (typically only used for passwords).
state: used to retrieve statistics and other info. State data is read-only.

Many data elements are scoped to a serial port (eg. a 4 serial port device will have 4 serial_setting elements). The port is identified as an attribute on the element of the form index=”num”. When a data element is specified without an explicit port attribute, the default is used. The default is “1”.

Example showing baud rate on serial port 2:
<serial index=”2”>
<baud>300</baud>
</serial>

4.5 Errors and Warnings

Response documents may contain an element as a child of the command or data element that indicates the result of the request. More than one error or warnings may be present.

Error and Warning elements

error	An error occurred.
warning	Command executed, but a warning was issued.

<error>

Attributes:
id A numeric id specified by the parent element (the command or the data element). An error element id=”0” is equivalent to no error.

Child Elements:
<desc>
Optional. Text description of the error.

<hint>
Optional. Used to indicate to the client the source of the error. This will typically be set to the field name that the error.

Example:

```
<serial_setting>
  <error id=”3”>
        <hint>baud</hint>
        <desc>Value out of valid range.</desc>
  </error>
</serial_setting>
```

Errors are required to have an id. <hint> and <desc> are optional and more than one are allowed.

4.6 Notes

4.6.1 RCI XML must be well-formed XML

The device parses incoming RCI requests in a sequential manner. Each XML element is parsed and acted upon as it arrives. This is not ideal behavior, but is necessary because of the inherent resource limitations of a device. Ideally, the entire XML request would be read into memory, validated, parsed and acted upon only after validation.

XML structure errors may be found after actions have been taken. For instance:

```
<rci_request version="1.0">
   <set_factory_default/>
</rci_requestBADENDTAG>
```

This request will result in an XML parse error, but since the parse error occurs after the set_factory_defaults, the device will be set to factory defaults.

Therefore, it is highly recommended that RCI requests be validated with an XML parser before being sent to the device. Using any standard parsers, such as the XML parsing in the Java SDK, to form RCI requests accomplishes this.

4.6.2 XML structure characters must not be sent as character data

Care must be taken to avoid accidental badly formed XML in RCI requests because of including XML structure characters, such as "<", as user entered data. Any field that accepts character data must be checked to ensure that "<" and ">" are not present (fields such as the email body of an alarm are common places this can happen).

It is recommended that all instances of "<" and ">" in character data be converted to "&lt" and "&gt", which is the standard XML representation of these characters.

5 RCI Data Elements

Setting

Boot Setting Group

RCI name: "boot"
Valid with: query_setting, set_setting, set_factory_default
No attributes supported.

A reboot is required for changes to take effect for boot settings.

Fields

Description	RCI element	r/rw	Values	Comments
DHCP (enable/disable)	dhcp	rw	on, off	Reboot required
IP Address	ip	rw	valid ip address	Reboot required
Subnet Mask	subnet	rw	valid ip address	Reboot required
Default Gateway	gateway	rw	valid ip address	Reboot required

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	6	Invalid DHCP setting
E	7	Invalid IP address
E	8	Invalid subnet mask
E	9	Invalid gateway address
E	13	ip address cannot start with 224-239 (class D network)
E	14	ip address cannot start with 240-255 (class E network)
E	15	Address cannot be the broadcast address
E	16	Invalid network configuration (network or host portion is invalid)
E	17	Invalid network configuration (no route from host to gateway)

Serial Setting Group

RCI name: "serial"
Valid with: query_setting, set_setting, set_factory_default
Attribute "index" specifies the serial port. Default is 1.

Fields

Description	RCI element	r/rw	Values	Comments
Baud Rate	baud	rw	50, 75, 110, 134, 150, 200, 300, 600,1200, 1800, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 230400	
Data Bits	databits	rw	5, 6, 7, 8	
Stop Bits	stopbits	rw	1, 2	
Parity	parity	rw	none, odd, even, mark, space	
Flow Control	flowcontrol	rw	none, software, hardware	
Port description	desc	rw	max 32 characters	

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid baud rate
E	5	Invalid databits
E	6	Invalid stopbits
E	7	Invalid parity
E	8	Invalid flow control
E	10	Invalid description

GPIO Mode Setting Group

RCI name: "gpio_mode"
Valid with: query_setting, set_setting, set_factory_default
No attributes.

Fields

Description	RCI element	r/rw	Values	Comments
GPIO Pin Mode	pin<x>	rw	serial, in, out, invalid	x is 1 to 5. Invalid returned on internal error. Pin cannot be set to "invalid". Ex: <pin1>in</pin1>

Errors and Warnings

E/W	ID	Description

E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Invalid pin
E	4	Invalid mode

Alarms Setting Group

RCI name: "gpio_alarm"
Valid with: query_setting, set_setting, set_factory_default

There are 32 configurable alarms which are independently configured below as indicated by the alarm element.

Fields

Description	RCI element	r/rw	Values	Comments
GPIO Alarm Service (enabled/disabled)	state	rw	on, off	
SMTP mail server used to send email	smtp_server	rw	valid ip address	
From: originator of email	from	rw	Max 64 characters	
Alarms 1 to 32	alarm	rw	n/a	Use attribute index to specify which alarm, 1 to 32.
Alarms Up to 32 alarms may be configured. Use elements of the form <alarm index="xx"> to configure alarms 1 through 32. The following elements are sub-elements of alarm.				
Turns on/off this alarm	state	rw	on, off	
Trigger on (transition/state)	trigger_on_state	rw	on, off	
Alarm Interval Time	alarm_interval	rw		Seconds. If state trigger, this is the interval to wait between sending alarms if trigger state achieved and is
Alarm Throttle Time	alarm_throttle	rw		Seconds, Alarms caused by transitions can be sent only once every alarm_throttle
GPIO Monitor Mask	monitor_mask	rw	4 byte hex mask: ex: 0x00000000	Identifies the GPIO pins that will be monitored for this trigger. This is a bit mask field, with the least significant 5 bits representing the GPIO lines. GPIO pin 1 is the least significant bit. The rest of the mask must be set to
GPIO State Mask	state_mask	rw	4 byte hex mask: ex: 0x00000000	Identifies the GPIO state (asserted/unasserted) to trigger on. This is a bit mask field, and the state bits correspond to the monitor mask.
Priority	priority	rw	normal, high	
Subject in email	subject	rw		
to: in email	to	rw	Max 64 characters	
cc: in email	cc	rw	Max 64 characters	
Serial data match string	match_string	rw		valid when trigger_mode is pattern_match
Trigger on gpio line or serial pattern match	trigger_mode	rw	gpio, pattern_match	

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid alarm index
E	5	Invalid global alarm state
E	6	Recipient too long
E	7	Invalid recipient
E	8	CC too long
E	9	Invalid CC
E	10	Sender too long
E	11	Invalid sender
E	12	Subject too long
E	13	Invalid subject
E	14	Invalid priority
E	15	Invalid smtp address
E	16	Invalid alarm interval
E	17	Invalid throttle interval
E	18	Invalid monitor mask
E	19	Invalid state mask
E	20	Invalid match string

E	21	Match string too long
E	22	Invalid trigger mode

Autoconnect Setting Group

RCI name: "autoconnect"
Valid with: query_setting, set_setting, set_factory_default
Attribute "index" specifies the serial port. Default is 1.

Autoconnect specifies the behavior of the Digi device when the Digi device initiates a TCP connection to a TCP server.

Fields

Description	RCI element	r/rw	Values	Comments
Autoconnect service (enable/disable)	state	rw	on, off	
Connect trigger	trigger	rw	always, data, dsr, dcd, string	
Connection type	service	rw	raw, ssl, telnet	
Description	desc	rw	Max 32 characters	
Destination to connect to	address	rw	valid IP address	
Socket on destination to connect to	port	rw		
Strip connect pattern from data sent	strip_pattern	rw	on, off	Not used unless trigger = string
Connect when this pattern is detected on the serial port	pattern	rw		Not used unless trigger = string

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid alarm index
E	5	Read only field
E	6	Invalid state
E	7	Invalid connect trigger
E	8	Invalid connect service
E	9	Invalid destination description
E	10	Invalid destination address
E	11	Invalid destination port
E	12	Invalid connect string

UDP Serial Setting Group

RCI name: "udp_serial"
Valid with: query_setting, set_setting, set_factory_default
Attribute "index" specifies the serial port. Default is 1.

There may be up to 10 UDP destinations.

Description	RCI element	r/rw	Values	Comments
UDP client service (enable/disable)	state	rw	on, off	
Send when pattern matched?	trigger_on_pattern	rw	on, off	
Strip patternn string before sending?	strip_pattern	rw	on, off	
Send after idle timeout?	trigger_on_timeout	rw	on, off	
Idle timeout	timeout	rw		Milliseconds
Send data threshold	count	rw		
Trigger pattern	pattern	rw	Max 16 characters	
Include socket id on send?	socketid_state	rw	on, off	
Socket id text to be included with data sent	socketid_string	rw		

UDP Destinations
Destinations 1 through 64 may be configured using the RCI element:
<dest index="xx"> where xx is 1 through 64.

The following elements are configured as sub-elements of <dest>

UDP Destination (enable/disable)	state	rw	on, off	
UDP Destination IP adddress	address	rw	valid IP address	
UDP Destination socket to connect to	port	rw		

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid state
E	5	Invalid trigger on pattern state
E	6	Invalid pattern
E	7	Invalid strip pattern state
E	8	Invalid trigger on timeout state
E	9	Invalid timeout value
E	12	Invalid destination state
E	13	Invalid destination description
E	14	Invalid destrination address
E	15	Invalid destination network port
E	16	Invalid socket ID

Inbound Network Services

A collection of settings groups that allow network services state(enabled/disabled) and port to be configured.
Valid with: query_setting, set_setting, set_factory_default
No attributes defined.

All of the following settings groups support the following fields: state, port, and desc.
state may be on or off.
port may be any valid network port not currently assigned to another service. Note, a port may be assigned to one UDP service and one TCP service.
desc is a description of the service and is read-only.

RCI Setting Group	Service Description	Default port	Comments
https	HTTPS	443	
secure_realport	Encrypted RealPort	1027	
telnet_server	Telnet Server	2001	
tcp_server	TCP Server	2101	
udp_server	Serial/UDP Server	2101	
securesocket	Secure Socket (SSL)	2601	
realport	RealPort	771	
telnet	Telnet Service	23	
lpd	Line Printer Daemon	515	

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Field is read-only
E	7	Invalid state
E	8	Invalid port

Simple Password Setting Group

RCI name: "simple_password"
Valid with: query_setting, set_setting, set_factory_default
No attributes defined.

Fields

Description	RCI element	r/rw	Values	Comments
User name	username	rw	Max 16 characters	
Security mode	password_mode	rw	always_reject, always_accept,	**always_reject** disables this username
SNMP public community name	public_comm	rw	Max 16 characters	default is "public"

SNMP private community name	private _comm	rw	Max 16 characters	default is "private"

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid username
E	5	Invalid password
E	6	Invalid mode

TCP Serial Setting Group

RCI name: "tcp_serial"
Valid with: query_setting, set_setting, set_factory_default
No attributes defined.

Fields

Description	RCI element	r/rw	Values	Comments
Send socket id?	socketid_state	rw	on, off	
Socket id to send	socketid_string	rw		
Drop TCP connection when DCD goes low?	hangup_dcd	rw	on, off	
Drop TCP connection when	hangup_dsr	rw	on, off	
Drop TCP connection on	idle_timeout	rw	on, off	

Errors and Warnings

E/W	ID	Description
E	1	Internal error (load failed)
E	2	Internal error (save failed)
E	3	Field specified does not exist
E	4	Invalid state
E	5	Invalid socket ID
E	6	Invalid timeout

State

GPIO State

RCI element "gpio"
No attributes defined

RCI Field Name	Setting	Values	Comments
pin<n>	GPIO Pin State	asserted, unasserted, unknown	x is 1 to 5. Ex. <pin1>asserted</pin1> Unknown will only be returned if an internal error occurred.

Device Information

RCI element "dev_info"
No attributes defined

RCI Field Name	Setting	Values	Comments
mac	MAC Address		Mac address in the form of aa:bb:cc:dd:ee:ff
product	Model (aka: Product Name)		Ex. Digi Connect ME
company	Manufacturer		Ex. Digi International Inc.
vendorid	Vendor ID		Ex. 44494749
boot	BOOT Version		Ex. Release_82000869_A
post	POST Version		Ex. Release_82000867_A
firmware	Firmware Version		Ex. Version 8200869_A 07/31/2003 16:45:34 CD

Serial Statistics

RCI element "serial_stats"
Attribute "index" specifies the serial port. Default is "1".

RCI Field Name	Setting	Values	Comments
overrun_err	Overrun errors		
overflow_err	Overflow errors		
frame_err	Frame errors		
parity_err	Parity errors		
breaks	Breaks		
rx	Receive bytes		
tx	Transmit bytes		
dtr	Current state		
rts	Current state		
cts	Current state		
dsr	Current state		
dcd	Current state		

Network Statistics

RCI element "net_stats"
No attributes defined.

These values correspond to values returned by the SNMP network MIB defined in rfc 1213, which should be consulted for the definition of these values.

RCI Field Name	Setting	Values	Comments
ifmtu			
ifspeed			
ifstatus			
ifinoctets			
ifinucastpkts			
ifinmucastpkts			
ifindiscards			
ifinerrors			
ifinunknownprotos			
ifoutoctets			
ifoutucastpkts			
ifoutmcastpkts			
ifoutdiscards			

ifouterrors			
ifoutqlen			
ipforwarding			
ipdefaultttl			
ipinreceives			
ipinhdrerrors			
ipinaddrerrors			
ipfowrwdatagrams			
ipinunknownprotos			
ipindiscards			
ipoutdiscards			
ipoutnoroutes			
ipreasmtimeout			
ipreasmreqds			
ipreasmoks			
ipreasmfails			
ipfragoks			
ipfragfails			
ipfragcreates			
iproutingdiscards			
icmpinmsgs			
icmpinerrors			
icmpindestunreachs			
tcptoalgorithm			
tcptoalgorithm			
tcptomin			
tcptomax			
tcpmaxconn			
tcpactiveopens			
tcppassiveopens			
tcpattemptfails			
tcpestabresets			
tcpcurrestab			
tcpinsegs			
tcpoutsegs			
tcpretranssegs			
tcpinerrs			
tcpoutrsts			
udpindatagrams			
udpnoports			
updinerrors			
udpoutdatagrams			

Device Statistics

RCI element "device_stats"
No attributes defined

RCI Field Name	Setting	Values	Comments
uptime	Up Time		Seconds
totalmem	Memory total		Bytes
usedmem	Memory used		Bytes
freemem	Memory free		Bytes

Boot Statistics

RCI element "boot_stats"
No attributes defined

RCI Field Name	Setting	Values	Comments
dhcp	DHCP (enable/disable)	on, off	
ip	IP Address		
subnet	Subnet Mask		
gateway	Default Gateway		