ConnectPort X2e ZB
User's Guide

ConnectPort X2e ZB Ethernet
ConnectPort X2e ZB Wi-Fi

90001298_A

©Digi International Inc. 2012. All Rights Reserved.

The Digi logo, iDigi, ConnectPort, are trademarks or registered trademarks of Digi International, Inc.

All other trademarks mentioned in this document are the property of their respective owners.

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

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

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

Contents

Introduction

CHAPTER 1

About this guide

This guide introduces the hardware, firmware, and software features of the ConnectPort X2e ZB. It describes how to perform configuration and administrative tasks, and how to develop and run applications on the device. Programming focuses on basic programming concepts and teaching through examples. More detailed programming content and program examples are provided in the Digi ESP™ for Python development environment. It also includes troubleshooting information, hardware specifications, certifications, and regulatory information. This guide is intended for a developer/programmer.

Digi contact information

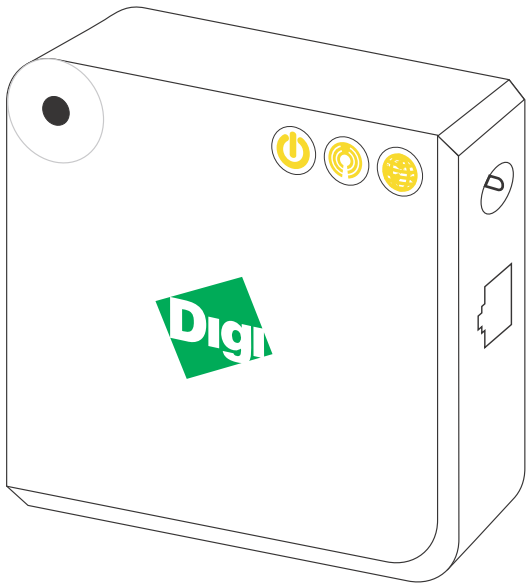
For more information about Digi products, or for customer service and technical support, contact Digi International.

To Contact Digi International by:	Use:
Mail	Digi International 11001 Bren Road East Minnetonka, MN 55343 U.S.A.
World Wide Web:	http://www.digi.com/support/
email	Look for the link **Contact Digi Support** at this address: http://www.digi.com/support/
Telephone (U.S.)	(952) 912-3444 or (877) 912-3444
Telephone (other locations)	+1 (952) 912-3444 or (877) 912-3444

ConnectPort X2e ZB

The ConnectPort X2e ZB is a compact, ZigBee®-to-IP gateway that provides low-cost IP connectivity of RF devices and sensor networks. Featuring an easy development environment, the ConnectPort X2e enables custom applications to run on the device, with access to XBee® or ZigBee wireless networks through an XBee RF interface, while providing WAN connectivity to cloud-based applications through Ethernet or Wi-Fi interface.

The ConnectPort X2e ZB features an end-to-end development environment, Digi ESP™ for Python, that allows for rapid M2M-specific application development on the industry standard Python scripting engine. The Digi ESP for Python environment is an IDE featuring device detection, debugging, compiling, and downloading of code to Digi gateways.



Hardware interfaces

ConnectPort X2e ZB hardware interfaces include a button for controlling various device operations, LEDs that indicate device state and status of connections, link and activity for Ethernet, Wi-Fi, and XBee network connections. Some of these hardware features can be controlled through programming. Hardware interfaces are covered in "Hardware" on page 14.

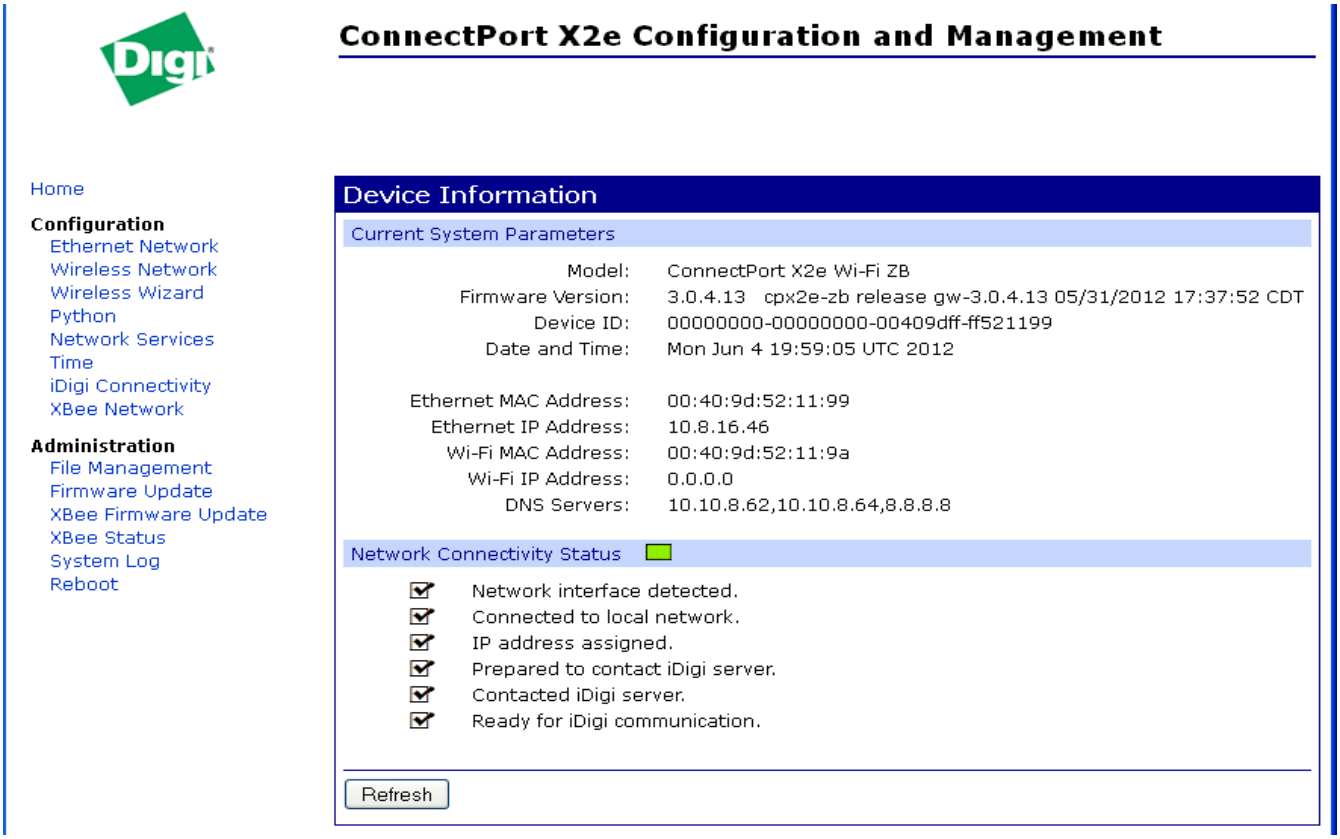
Configuration and management interfaces

To establish network connectivity with a ConnectPort X2e ZB device, minimal configuration is required in many environments. This means that you may not need to set or change configuration settings from their factory defaults to begin developing with the device. As necessary, there are several user interfaces for interacting with the ConnectPort X2e, for example, to view or change configuration settings or perform important administrative tasks such as updating firmware or rebooting the device.

- A web-based interface for configuring, monitoring, and administering Digi devices.
- iDigi Manager Pro, a remote-management interface.

Web interface

A web interface is provided as an easy way to configure basic device settings and perform administrative tasks. This interface is available via a local network connection to the ConnectPort X2e ZB. Configurable features are grouped into several categories.



iDigi Manager Pro™ interface

The default behavior of Connectport X2e ZB is to power up and connect to the iDigi® Device Cloud™. iDigi Manager Pro is a Digi-hosted remote management interface within the iDigi Device Cloud. It offers a platform for secure, scalable access to an unlimited number of remote assets. The iDigi Manager Pro interface allows you to:

- Configure devices
- Remotely update device firmware
- Upload and manage Python/iDigi Dia files
- Remotely reboot devices
- Reset devices to factory defaults
- Import or export the device configuration properties
- Track devices
- Monitor devices and connections.

In addition, the iDigi web services provide seamless integration from Digi gateways into customer back office applications.For more information on iDigi Manager Pro as a remote management interface, see these resources:

- *iDigi User's Guide*, available on digi.com
- *iDigi Web Services Programming Guide*, available on digi.com
- iDigi tutorials and other documents available on iDigi.com

Programming interfaces and capabilities

The ConnectPort X2e ZB allows custom embedded logic via the Python scripting language. To meet the needs of customers with varying levels of Python expertise and application complexity, a number of development strategies are supported, which can be mixed and matched as a developer sees fit:

Python development independent of the target device

The ConnectPort X2e ZB features a standard Python 2.7 distribution, allowing applications that are not dependent on Digi-proprietary interface modules to be developed and tested independently of the device. Scripts developed in this manner can generally be transferred to the device for final testing at the end of the development cycle, with a PC serving as a device proxy during the bulk of development.

Python development on the target device

The ConnectPort X2e ZB features a Linux shell interface allowing a developer to experiment with the Python interpreter interactively, create scripts, launch scripts, and control their operation.

Digi ESP for Python

The ConnectPort X2e ZB is supported by the Digi ESP for Python, an IDE featuring device detection, debugging, compiling and downloading of iDigi Dia/Python code to Digi gateways. Integrated into the Digi ESP are example applications that can demonstrate the use of some of Digi's proprietary Python extensions, serving as templates for applications seeking to incorporate common functionality.

iDigi Dia

The ConnectPort X2e ZB is supported by the iDigi Dia application framework, which is provided within Digi ESP for Python. The iDigi Device Integration Application ("Dia") is an application software platform for Digi gateways. iDigi Dia makes connecting remote devices and sensors to Digi gateway products easy by providing a ready-to-use software. Put simply, the iDigi Dia framework is used to gather data from XBee sensor networks, transform the data into a useful form, and push the data to the iDigi Device Cloud for consumption by a user. iDigi Dia is written in the Python programming language, and can be extended to meet unique device connectivity requirements. Particularly valuable in conjunction with the Digi ESP for Python, the iDigi Dia framework seeks to shrink the development cycle for complex data gathering and transformation applications. With abstractions for components like interface handling (drivers), data management (channels), data delivery (presentations); and a library of ready-to-use modules for common operations; the iDigi Dia framework allows a developer to focus efforts on proprietary logic, not the glue that holds an application together. The iDigi Dia application framework also provides the shortest path to integration on proprietary logic with Digi's network device and data management platform, iDigi.

Linux command shell

Note: The ConnectPort X2e ZB also has a Linux shell command-line interface. While Digi ESP for Python is intended as the main programming interface, this interface may be used for some programming and device management tasks. This interface is accessed with a username of **python** and password **dbps**, and is described in "The Linux command shell (command-line interface)" on page 68.

Product differences from predecessor ConnectPort X products

ConnectPort X2e ZB gateways differ from predecessor ConnectPort X products. These differences are of importance to programmers and integrators who are familiar with the predecessor devices and need to develop applications and install or manage the gateway. This list of differences assumes that the reader has knowledge of the features and functions of predecessor ConnectPort X products.

- **Operating system**: The ConnectPort X2e ZB is built on the industry-standard Linux operating system, versus a Digi-proprietary embedded operating system.
- **Memory**: The ConnectPort X2e ZB has 64MB of RAM and 128MB of Flash memory. Users have access to up to 20MB of RAM and up to 20MB of Flash memory. Predecessor devices had less RAM and Flash available for custom Python applications.
- **System date and time**: The ConnectPort X2e ZB, for reasons of improved security, has a greater dependence on time synchronization than predecessor products. In so doing, the ConnectPort X2e ZB expects to use the standard Network Time Protocol (NTP), requiring connectivity with an external NTP time server. Without a proper sense of time, the device will be unable to correctly validate security certificates, disabling the ability to connect to iDigi, as well as disabling the ability to update the firmware.
- **Button**: The ConnectPort X2e ZB features a programmable button. The button can be configured to activate some Digi native features (such as returning a device to its factory defaults), and can also be used by custom applications. For more information on the button see page 20. This button behavior differs from the Reset button behavior on the ConnectPort X2.
- **LED behaviors and meanings**: The ConnectPort X2e ZB has some differences in LED behavior and meaning from predecessor ConnectPort X products. See page 17 for descriptions and page 62 for programming example.

- **User interfaces**:
  - The ConnectPort X2e ZB has a web user interface for both network configuration and access to the log file for troubleshooting the initial connection to iDigi. For more information on the web interface, see "Configuration" on page 27.
  - Command-line interface differences: The ConnectPort X2e ZB allows access to the Linux shell using SSH. This shell and common operations are described on page 68. Access to the gateway is at the user level; the user is named **python** and the password is **dbps**. Access to a command-line interface through Telnet is not supported for network security reasons. Commands in the command-line interface for predecessor ConnectPort X products are not supported. However, some of the ConnectPort X2 command-line interface commands have equivalents in the Remote Command Interface (RCI).
- **Firmware updates**: Due to the complexity of the Linux-based system, standard firmware updates cannot be used to downgrade a system.
- **Logging**: The ConnectPort X2e ZB supports continuous logging for troubleshooting. The log files can be browsed from the web interface or pulled from the device filesystem in the iDigi interface. They are stored in the Linux filesystem in the /WEB/logging directory and persist across reboots and power cycles.
- **Supported Python version**: The ConnectPort X2e ZB uses Python interpreter version 2.7. Many predecessor ConnectPort X products use Python 2.4.
  Any custom-compiled Python code must be recompiled for Python interpreter 2.7. Custom Python modules are not 100% compatible with the ConnectPort X2e ZB. Therefore, porting may be required in addition to recompiling.

Where to find more information

In addition to this guide, find additional product and feature information in the these documents. These documents are available from Digi's Support page http://www.digi.com/support/ unless otherwise noted.

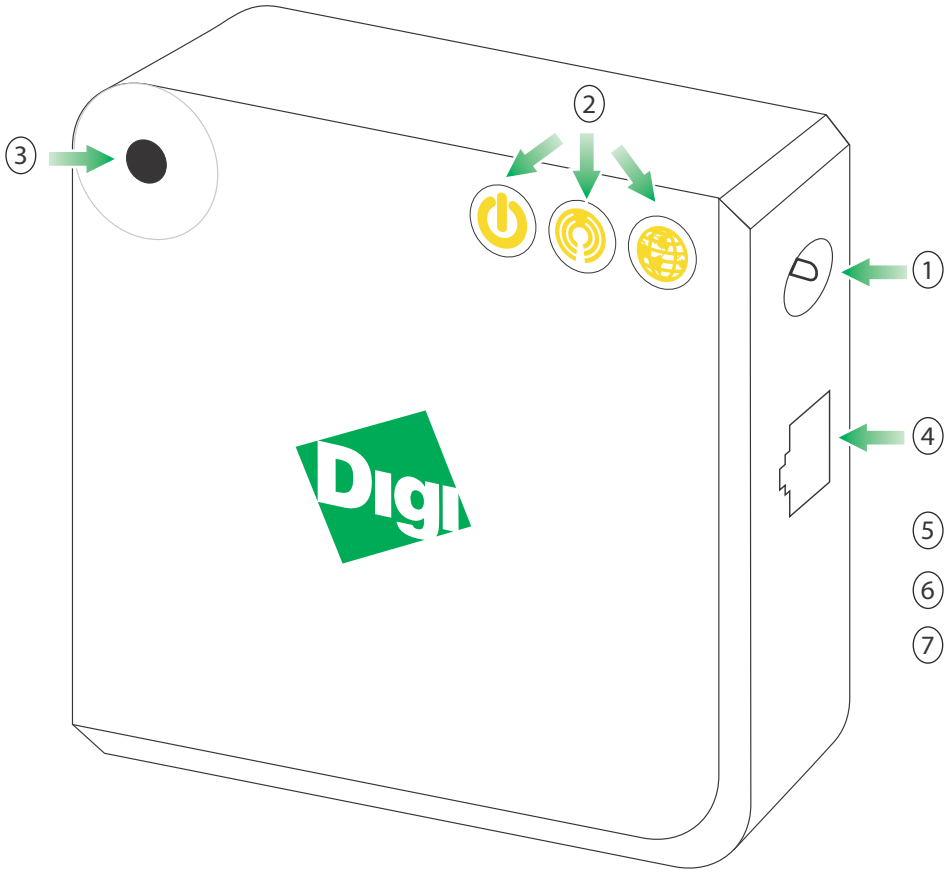
- *XBee/XBee-PRO ZB SMT RF Modules Product Manual* (90000976), for more information about features and operation of the XBee RF module mounted inside the gateway.
- ConnectPort X2e ZB programming content within Digi ESP for Python
- The Wiki page for additional programming content for ConnectPort X2e products:
  http://www.digi.com/wiki/developer/index.php/ConnectPort_X2e
- The Programming section lists several references and tutorials for users new to a Linux-based filesystem and the ash command shell.iDigi Dia library documentation
- *iDigi User's Guide*
- *iDigi Web Services Programming Guide*
- iDigi tutorials and other documents available on iDigi.com

Hardware

CHAPTER 2

This section provides hardware specifications, reviews key hardware features, and lists regulatory statements and certifications of the ConnectPort X2e ZB.

Hardware feature summary



1 Power requirements. See page 15.
2 LED status indicators. See page 17.
3 Button. See page 20.
4 Ethernet features. See page 15.
5 Wi-Fi features. See page 16.
6 XBee RF module. See page 16 and "XBee LED" on page 18.
7 Antennas (internal). See page 21.

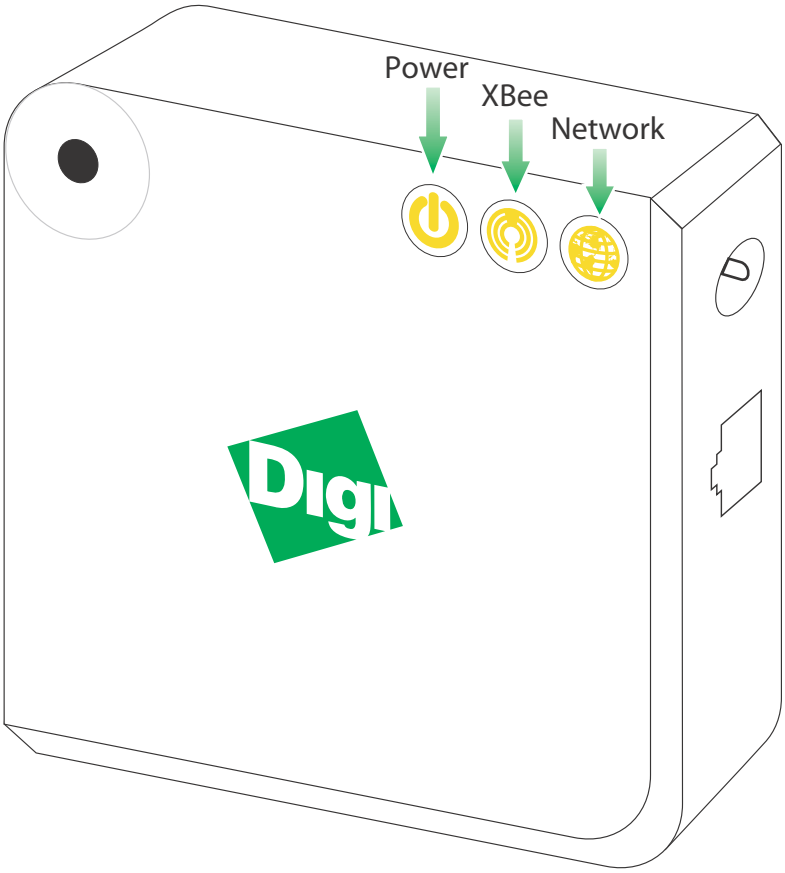
Hardware specifications

Specification		Value
Environmental	Operating temperature	32° F to 104° F (0° C to 40° C)
	Relative humidity	5% to 95% (non-condensing)
	Storage and transport temperature	-40 to 185F (-40 to 85C)
	Altitude	6560 feet (2000 meters)
	Ethernet isolation	500 VAC min per IEEE802.3/ANSI X3.263
Power requirements	DC power input	■ Voltage input: 5 VDC +/- 5% ■ Power consumption: Typical: 1.2 W, Max: 2.5 W ■ Connector: 2.35mm x 5.7mm, center pin positive.
Dimensions	Length	3 in (7.62 cm)
	Width	3 in (7.62 cm)
	Depth	1 in (2.54 cm)
	Weight	0.15 lb (0.07 kg)
Ethernet	Ethernet Ports	1 RJ-45 port
	Physical Layer	10/100 Base-T (Auto-MDIX)
	Data Rate	10/100 Mbps (auto-sensing)
	Mode	Full or half duplex (auto-sensing)

Specification		Value
Wi-Fi	802.11	b/g/n (2.4GHz only)
	Data Rate	Up to 72.2 Mbps
	Transmit Power	18 dBm typical (varies by mode and channel)
	Receiver Sensitivity	-87 dBm @ 11 Mbps
	Modes	Infrastructure Client mode only
XBee See the *XBee/XBee-PRO ZB SMT RF Modules Product Manual* (90000976) for complete specifications and product information.	Module type	XBee® ZB SMT
	Transmit power	■ The Domestic product version uses the XBee-PRO RF module, with a transmit power of 63mW (+18dBm). ■ The International product version uses the regular XBee RF module, with a transmit power of 6.3 mW (+8 dBm).
	Receiver sensitivity (1% PER)	-102 dBm
Development	Python version	2.7.1
	Memory	64 MB RAM, 128 MB Flash 20 MB RAM 20 MB Flash available for Python applications

LEDs

The ConnectPort X2e ZB has several LEDs. All LEDs have default behaviors, described in this section. Some LEDs can also be controlled programmatically, as discussed in "LED control" on page 62.



Power LED

Color and blink pattern	Description
Off	No power.
Solid green	Device is powered. This state does not indicate that the device is fully operational. The Network LED and its states convey such information.

XBee LED

The XBee LED indicates the status of the connection of the XBee RF module in the ConnectPort X2e ZB to an XBee wireless network. The behavior of the XBee LED varies depending on whether the ConnectPort X2e ZB acts as a coordinator or a router. For information on changing the ConnectPort X2e ZB from a coordinator to a router, see the *XBee/XBee-PRO ZB SMT RF Modules Product Manual*.

ConnectPort X2e ZB as Coordinator

Color and blink pattern	Description
Solid green	The XBee RF module has not started a network.
Blinking green	The XBee RF module has started a network.

ConnectPort X2e ZB as Router

Color and blink pattern	Description
Solid green	The XBee RF module has not joined a network.
Blinking green	The XBee RF module has joined a network.

Network LED

The Network LED indicates the status of the connection of the ConnectPort X2e ZB to both a communications network and an iDigi server. The LED can be user-controlled; see page 62 for details.

Ethernet or Wi-Fi network communication

Color and blink pattern	Description
Off	Gateway operating system is not yet running.
Blinking yellow (slow)	Operating system is running, but no Ethernet or Wi-Fi link is established.
Blinking yellow (fast)	Ethernet or Wi-Fi link is established, but no IP address is assigned to the gateway.
Solid yellow	An IP address has been assigned to the gateway.
Alternating between yellow and green	The gateway cannot connect to iDigi. The two most common reasons for this state are: ■ The gateway cannot connect to a DNS server to resolve the iDigi server address. See page 72. ■ The gateway cannot connect to an NTP time server to get the correct time. See page 72.
Blinking green (slow)	Gateway is attempting a connection to an iDigi server.
Blinking green (fast)	iDigi server found and gateway is undergoing authentication.
Solid green	Gateway is connected to the iDigi server.

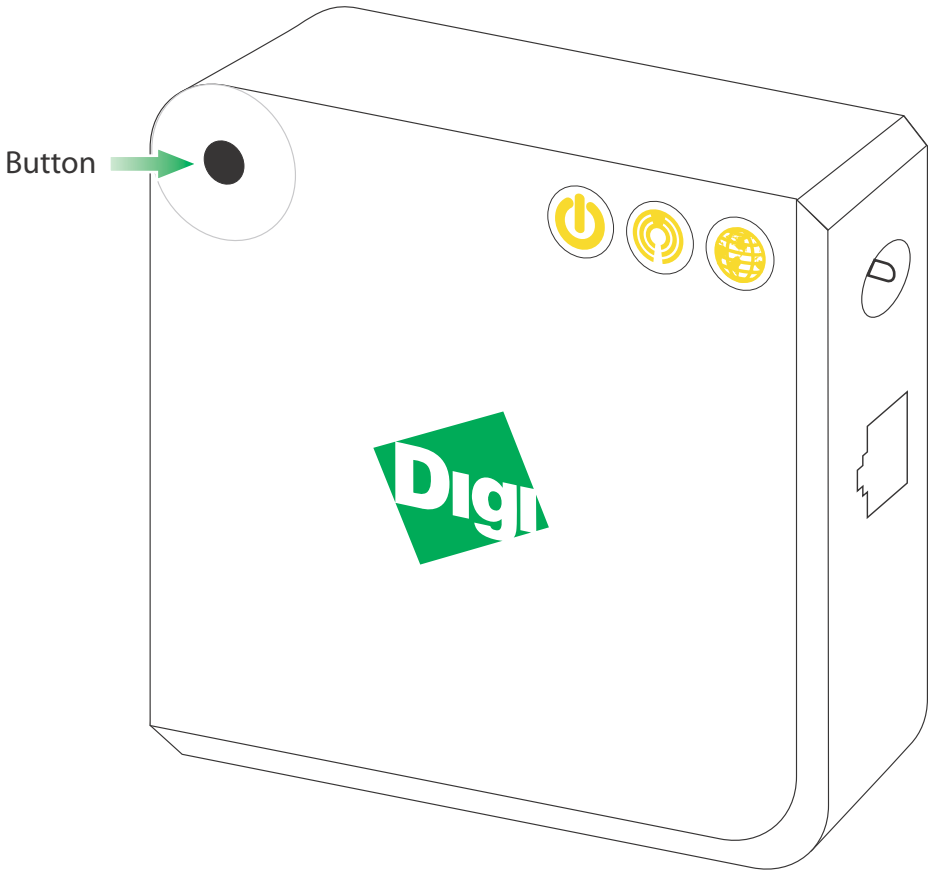
Button

The button on the ConnectPort X2e ZB is capable of performing a number of Digi-defined actions, including:

- Reset the device configuration settings to their factory defaults
- Enable configuration changes via the device discovery tools within a five-minute window
- Enable the device web interface
- Enable a special purpose Wi-Fi configuration mode

Some of these features are enabled on the button by default, but all behaviors can assigned either to the “default” button, or to “none” (to disable the feature); see below for detailed descriptions of these features. This assignment is done through the iDigi Manager Pro interface, in **Properties > Advanced Configuration > Button Service Assignment**. The choices for each button service assignment are “**Default button,” “per Digi product specification**,” which means to use the default behaviors described on the following page, and “**Feature disabled**,” which means that the feature is not assigned to the button.

The button state can also be read by a Python application. This can be done in conjunction with the Digi standard actions, the button behaviors can be enabled/disabled individually, or all Digi behaviors can be disabled to provide full responsibility for the button to an application. See "Button handling" on page 61 for details.



Factory reset

Pressing the button for over 10 seconds after the ConnectPort X2e ZB is running resets the device to its factory default configuration. This action clears any configuration settings you may have entered through the supported device interfaces. This feature is assigned to the button by default.

Enable configuration changes via the device discovery tools

The default behavior for the ConnectPort X2e ZB is that it can be discovered through Digi Device Discovery and configuration settings can be displayed and changed, with no time-limit window; therefore, the button behavior is disabled. This feature, when enabled, restricts device state changes via the device discovery tools to a five-minute window since the most-recent button press.

Note: When an attempt is made to change configuration settings using the Digi Device Discovery tool, a password prompt will be displayed. Leave the field in the prompt blank and click **OK**.

Enable the device web interface

The default behavior for the ConnectPort X2e ZB is that the feature of controlling whether the web interface is exposed is disabled. That is, the web interface is always available and no button press is needed to access the web interface. When this feature is assigned to the button, and web interface availability is restricted, a user needs to press the button to access the web interface. The web interface will be open for a five minute window since the most recent button press.

Enable a special-purpose Wi-Fi configuration mode

When assigned to the button, the feature of enabling a special-purpose Wi-Fi configuration mode allows a button press to create a temporary access point for configuration of the device *if* the device has not yet already been configured for Wi-Fi. Each button press extends the window of access point mode operation to five minutes from the time of the button press. If the feature is not assigned to the button, no Wi-Fi configuration access point mode is available. The default behavior on the ZB is the feature is assigned to the button.

When the special-purpose Wi-Fi configuration mode is enabled, enable Access Point mode on your device by pressing the button. Configure the Wi-Fi interface of your laptop to connect to the ConnectPort X2e ZB gateway's access point. The name (SSID) of the access point will be **cpx2e-zb-*xxxxxxxxxx***, where ***xxxxxxxxxx*** is the serial number of the gateway.

Antennas

The ConnectPort X2e ZB has internal antennas. All models have an internal antenna for the XBee RF module. Wi-Fi models have an additional internal antenna.

Regulatory information and certifications

RF exposure statement

In order to comply with RF exposure limits established in the ANSI C95.1 standards, the distance between the antenna or antennas and the user should not be less than 20 cm.

FCC certifications and regulatory information (USA only)

FCC Part 15 Class B

Radio Frequency Interface (RFI) (FCC 15.105)

This device has been tested and found to comply with the limits for Class B digital devices pursuant to Part 15 Subpart B, of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential environment. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try and correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Labeling Requirements (FCC 15.19)

This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

If the FCC ID is not visible when installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module FCC ID.

Modifications (FCC 15.21)

Changes or modifications to this equipment not expressly approved by Digi may void the user's authority to operate this equipment.

European Community - CE Mark Declaration of Conformity (DoC)

We:

Manufacturer's Name:	Digi International

of:

Corporate Headquarters:	11001 Bren Road East Minnetonka MN 55343
Manufacturing Headquarters:	10000 West 76th Street Eden Prairie MN 55344

Declare under our sole responsibility that the product:

Product Name	**Model Number**
ConnectPort X2e ZB	50X2E-Z3C-*XXX-XX*

to which this declaration relates are in conformity with the essential requirements and other relevant requirements of Directive 1999/5/EC (R&TTE):

- Safety (article 3.1a)
  - EN 60950-1:2006
- EMC (article 3.1b)
  - EN 55022:2010 Class B
  - EN 55024:2010
  - EN 61000-3-2:2006
  - EN 61000-3-3:2008
  - EN 301 489-17 V2.1.1:2009 Class B
- Spectrum (article 3.2)
  - EN 300 328 V1.7.1:2006

Industry Canada (IC) certifications

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class B prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

Safety statements

Important Safety Information

To avoid contact with electrical current:

- Never install electrical wiring during an electrical storm.
- Never install an Ethernet connection in wet locations unless that connector is specifically designed for wet locations.
- Use caution when installing or modifying lines.
- Use a screwdriver and other tools with insulated handles.
- Wear safety glasses or goggles.
- Do not place Ethernet wiring or connections in any conduit, outlet or junction box containing electrical wiring.
- Installation of inside wire may bring you close to electrical wire, conduit, terminals and other electrical facilities. Extreme caution must be used to avoid electrical shock from such facilities. Avoid contact with all such facilities.
- Ethernet wiring must be at least 6 feet from bare power wiring or lightning rods and associated wires, and at least 6 inches from other wire (antenna wires, doorbell wires, wires from transformers to neon signs), steam or hot water pipes, and heating ducts.
- Do not place an Ethernet connection where it would allow a person to use an Ethernet device while in a bathtub, shower, swimming pool, or similar hazardous location.
- Protectors and grounding wire placed by the service provider must not be connected to, removed, or modified by the customer.
- Do not touch uninsulated Ethernet wiring if lightning is likely!
- External Wiring: Any *external* communications wiring installed needs to be constructed to all relevant electrical codes. In the United States this is the National Electrical Code Article 800. Contact a licensed electrician for details.

5.10 Ignition of Flammable Atmospheres

Warnings for Use of Wireless Devices

Observe all warning notices regarding use of wireless devices.

Potentially Hazardous Atmospheres

Observe restrictions on the use of radio devices in fuel depots, chemical plants, etc. and areas where the air contains chemicals or particles, such as grain, dust, or metal powders, and any other area where you would normally be advised to turn off your vehicle engine.

Safety in Aircraft

Switch off the wireless device when instructed to do so by airport or airline staff. If the device offers a 'flight mode' or similar feature, consult airline staff about its use in flight.

Safety in Hospitals

Wireless devices transmit radio frequency energy and may affect medical electrical equipment. Switch off wireless devices wherever requested to do so in hospitals, clinics, or health care facilities. These requests are designed to prevent possible interference with sensitive medical equipment.

Pacemakers

Pacemaker manufacturers recommended that a minimum of 15cm (6 inches) be maintained between a handheld wireless device and a pacemaker to avoid potential interference with the pacemaker. These recommendations are consistent with independent research and recommendations by Wireless Technology Research.

Persons with Pacemakers:

- Should ALWAYS keep the device more than 15cm (6 inches) from their pacemaker when turned ON.
- Should not carry the device in a breast pocket.
- If you have any reason to suspect that the interference is taking place, turn OFF your device.

International EMC (Electromagnetic Emissions/Immunity/Safety) standards

This product complies with the requirements of following Electromagnetic Emissions/Immunity/Safety standards.

There are no user-serviceable parts inside the product. Contact your Digi representative through "Digi contact information" on page 8 for repair information.

Emissions	Immunity	Safety
AS/NZS 4268:2008 (Amended by A1:2010) Class B (Wi-Fi only) AS/NZS CISPR 22:2009 Class B EN 301 489-17 V2.1.1:2009 Class B (Wi-Fi only) EN 55022:2010 Class B EN 61000-3-2:2006 EN 61000-3-3:2008 FCC Part 15 Subpart B Class B FCC Part 15 Subpart C (Wi-Fi only) ICES-003:2004 Class B RSS-Gen:2010 (Wi-Fi only) RSS-210:2010 (Wi-Fi only)	EN 304 489-17 V2.1.1:2009 (Wi-Fi only) EN 55024:2010	IEC 60950-1:2005 EN 60950-1:2006 UL 60950-1 CSA C22.2 No. 60950-1

Configuration

CHAPTER 3

While the ConnectPort X2e ZB is designed to allow network communication with minimal configuration, there are several configuration settings that can be adjusted. This section covers those configuration settings and configuration of these settings from the web interface, and from iDigi Manager Pro.

Note: configuration can be performed programmatically

In addition to the methods described in this chapter, configuration can be performed programmatically, through iDigi Web Services, and natively using Python modules. See page 58.

Important configurable settings

For the ConnectPort X2e ZB, most settings have reasonable defaults that do not need to be changed. However, these are some important settings available in the configuration interfaces that get the device up and communicating.

- Ethernet IP network settings
- Wi-Fi settings
- NTP server settings
- iDigi remote device management

Ethernet IP network settings

Ethernet IP network settings configure how the IP address of the Digi device for Ethernet network communications is obtained, either by DHCP or by using a static IP address, subnet mask, and default gateway, and Domain Name System (DNS) servers.

DHCP

The ConnectPort X2e ZB uses a DHCP server to obtain its IP address information, by default. A DHCP server needs to provide an IP address, subnet mask, default gateway, and Domain Name System (DNS) server for the device. If you disable DHCP, you must set all of these settings yourself.

In the absence of a DHCP server, a static IP address will need to be assigned, most easily by accessing the device Digi Device Discovery tool and changing the network settings through that interface.

IP Address
Subnet Mask
Default Gateway

These settings should be considered as a group, and only come into play when automatic IP address assignment is disabled.

- **IP Address**: The IP address is a 4-part ID assigned to network devices. IP addresses are in the form of 192.168.2.2, where each number is between 0 and 255.
- **Subnet Mask**: The Subnet Mask is combined with the IP address to determine which network this Digi device is part of. A common subnet mask is 255.255.255.0.
- **Default Gateway**: The IP address of the computer that enables this Digi device to access other networks, such as the Internet.

DNS servers

A DNS (Domain Name System) server is an Internet service that resolves domain names into IP addresses. Name resolution is important when connecting to iDigi, as the Digi servers are provided as fully-qualified domain names.

ConnectPort X2e ZB is capable of using up to three DNS servers. Up to two of these slots may be filled with DNS servers from dynamic IP assignment sources, leaving at least one slot always available for static DNS server configuration. A reasonable default is supplied for one static DNS server, but this default may not be appropriate for all customer networks.

Wi-Fi settings

If the ConnectPort X2e ZB will use Wi-Fi communications, several Wi-Fi settings are important to enable network connectivity.

- A **Wi-Fi configuration wizard** is used to configure the Wi-Fi interface. This wizard "teaches" a Wi-Fi ConnectPort X2e ZB device the wireless parameters needed to further configure wireless settings and operation.
- **SSID**: The name of the Wi-Fi network to connect to. enable Access Point mode on your device by pressing the button. Configure the Wi-Fi interface of your laptop to connect to the ConnectPort X2e ZB gateway's access point. The name (SSID) of the access point will be **cpx2e-zb-*xxxxxxxxxx***, where ***xxxxxxxxxxx*** is the serial number of the gateway.
- **Network security**: The type of network security used on the wireless network will need to be specified.

NTP server settings

The ConnectPort X2e ZB uses Network Time Protocol (NTP) for time synchronization. Using NTP requires an external NTP time server. Time synchronization is critical to the security of the device, including validating the certificates that sign firmware update images, as well as to verify the server certificate if/when connecting to iDigi. Steps are taken to preserve a sense of the time across reboots, but the availability of an NTP server or servers is important to the long-term health of the device.

The NTP servers can be configured in the web interface on the **Configuration > Time** page. In environments where the device cannot directly connect to the Internet, configuration changes to point to a local NTP server are important. Many users will not need to change the time server setting. If already connected to iDigi, it is also possible to adjust the time server configuration.

iDigi remote device management

The ConnectPort X2e ZB, as with many of its predecessors, is compatible with the iDigi device and data management platform. iDigi provides a mechanism to do more advanced device configuration than is generally possible in the web interface Once the device has established network connectivity to the iDigi server, it will be manageable remotely using the iDigi interface. Configurable iDigi settings include:

- **iDigi Connectivity**: Enabled by default. You may wish to disable this feature if you have no use for iDigi, and wish to eliminate any iDigi-related network traffic.
- **iDigi Server Name**: The value for this setting is generally a Fully Qualified Domain Name (FQDN) pointing to one of the Digi iDigi servers. This setting is configurable through the web interface, in the **Configuration > iDigi Connectivity** page, and through iDigi, in the Advanced configuration settings.
- **iDigi Server Port**: This setting is configurable on the iDigi Configuration page, not through iDigi Manager Pro.
- **Proxy Server Name / Port:** A path to an iDigi server can be made through a local HTTP proxy, such as squid. provided that server is used to simply remap the target IP address and port number, without extra authentication or other security measures.

Additional device configuration settings

There are several additional device features that are not essential for IP network connectivity, but which will require configuration settings to be entered or changed from their defaults.

Python

Any Python programs loaded onto the ConnectPort X2e ZB can be configured to start automatically at system startup.

XBee settings

The ConnectPort X2e ZB provides a gateway between an Internet Protocol (IP) network wired or wireless devices and a network of various wireless devices containing XBee or ZigBee wireless RF modules. Typically, these wireless devices are small sensors and controllers. Remote nodes in an XBee network can be other Digi ZB or SE nodes, or 3rd-party nodes.

The XBee module can be configured as a coordinator or router in an XBee or ZigBee network. For information on configuring the XBee module as a coordinator or router, and for complete XBee module settings and their descriptions, and discussions of XBee network concepts, see the *XBee/XBee-PRO ZB SMT RF Modules Product Manual* (90000976).

Configuration from the Web interface

The web interface for ConnectPort X2e ZB is intended to support basic feature configuration as well as critical network configuration. Not every device setting is displayed in this interface. For more extensive access to settings, use the iDigi Manager Pro interface.

Open the web interface

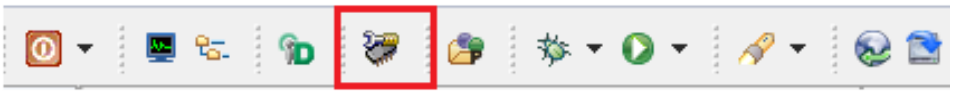
To open the web interface, use the Digi Device Discovery utility in Digi ESP for Python to get the IP address for the device. The Digi Device Discovery utility locates Digi devices on a network. It uses a Digi International-proprietary protocol, Advanced Digi Discovery Protocol (ADDP), to discover the Digi devices on a network, and displays the discovered devices in a list. Digi Device Discovery can be launched from the Digi ESP for Python interface.

Note: If you already know the IP address for the device, you can open a web browser and enter the IP address in the address bar to open the web interface. For example http://10.101.1.178

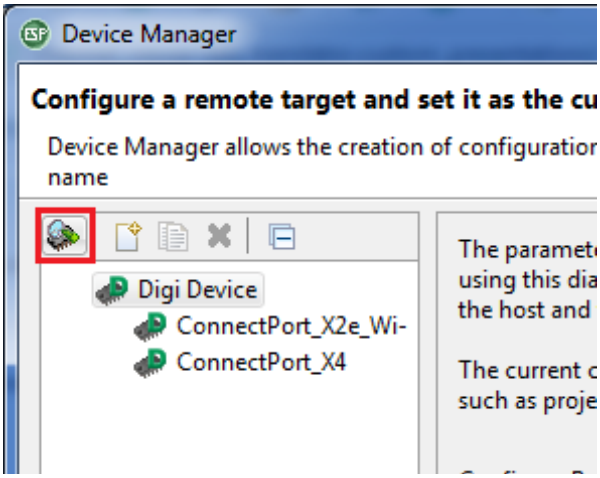
To launch Digi Device Discovery within Digi ESP:

1 Launch Digi ESP from the Start menu: **Start > Digi > Python > Dev Tools... > Digi ESP for Python.**

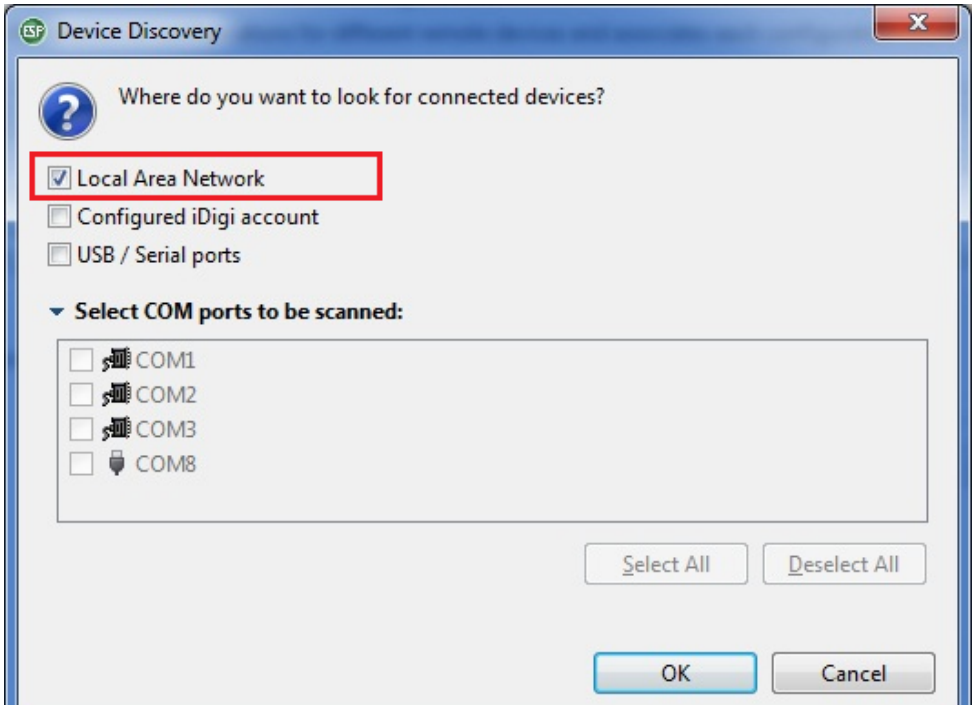
2 Open the Device Manager: from the Digi ESP main menu, selecting **Device Options > Device Manager**, or on the toolbar, click the ***Device Manager*** button.



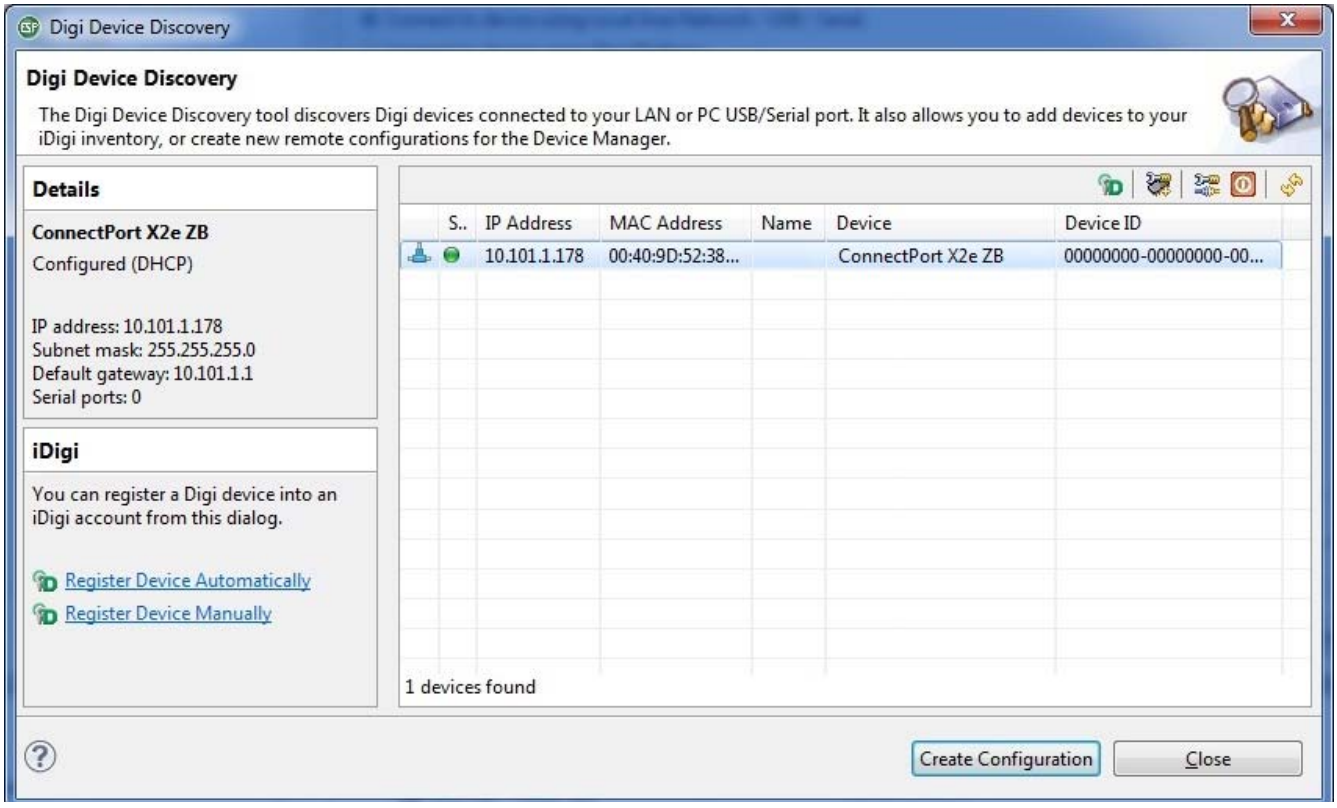
3 In the Device Manager window, click on the ***Device Discovery*** button located on the toolbar above the list of available remove devices.



4 A dialog is displayed for asking where to look for connected devices. Make sure the option **Local Area Network** is selected and click **OK**.



5 The **Digi Device Discovery** dialog is displayed, listing the devices found. The **IP Address** column displays the IP address of every device. Locate the device in the list of devices.



6 Open an internal browser within Digi ESP by choosing **Window > Show View > Other > Internal Web Browser**. Click **OK.**

7 In the browser, enter the IP address of your device, in the address bar of the browser:

http://*device_ip_address*

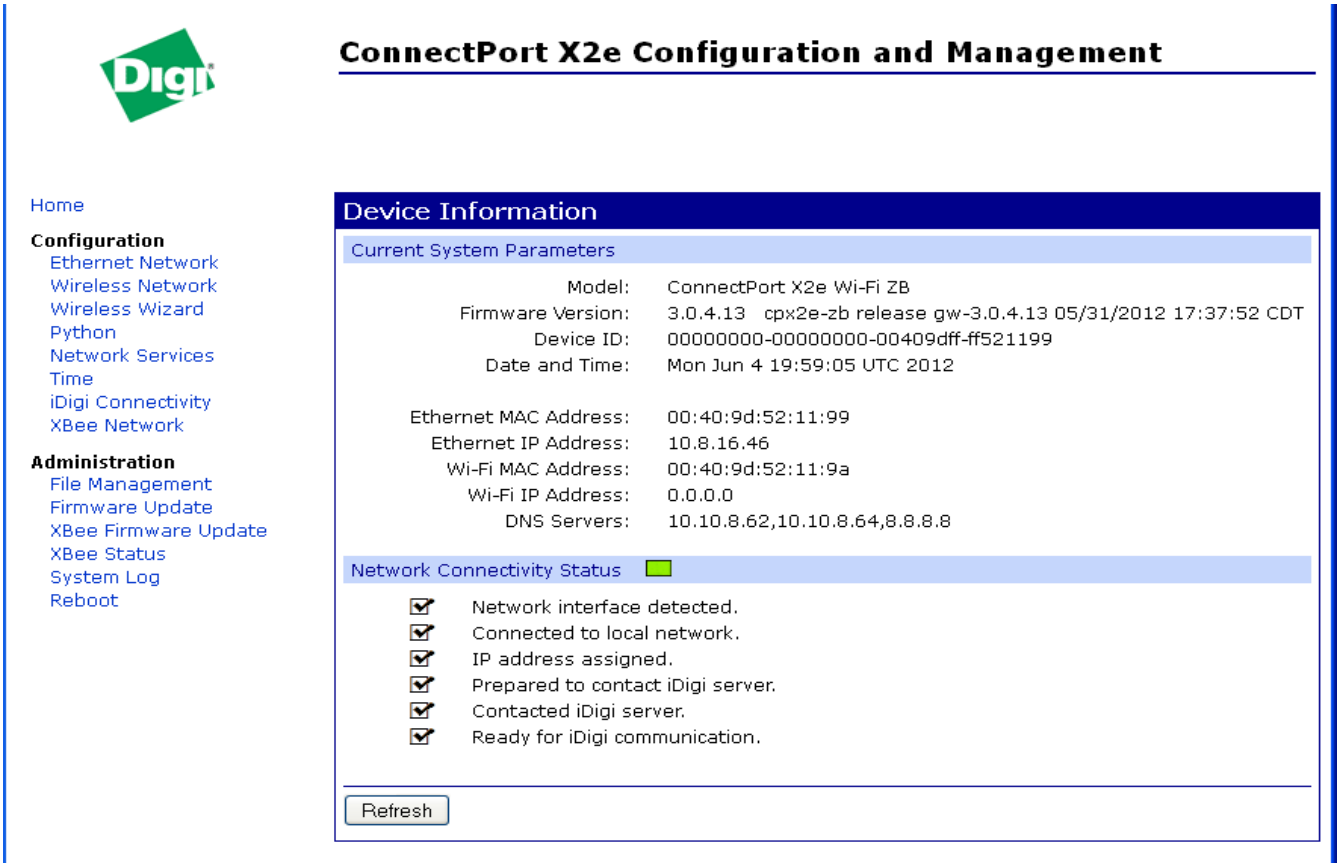
For example:

http://10.101.1.178

Note: Use of Digi Device Discovery within Digi ESP is covered further in the Digi ESP help topic **Help > Help Contents > Digi ESP for Python > 3. Working with the IDE > 3.3. Configuring Devices > 3.3.2. Automatic Configuration**.

The Home page

When the web interface is opened, the **Home** page is displayed. The information listed on this page may vary based on product and supported features.



Device Information

The **Device Information** section of the Home page summarizes current system parameters and network connectivity status.

The **Network Connectivity Status** LED and information displayed indicates the readiness of the ConnectPort X2e ZB to communicate in a network and with the iDigi server. See the description of the status LEDs beginning on page 17 and "Troubleshooting" on page 69 for information on the LED and the various network connectivity status conditions listed.

Refresh button

Clicking **Refresh** refreshes the Home page. This refresh operation is necessary because things like system time and network connectivity status are not dynamically updated when the state changes on the device. This refresh operation also updates device status information.

Configuration and Administration links

The left side of the Home page has a menu of choices that display pages for configuration and administration tasks.

The choices under **Configuration** in the menu display pages for configuring settings for various features. Some of the configuration settings are organized on sets of linked screens. The choices in this menu may vary based on product and supported features.

The choices under **Administration** complete common device administration tasks, and are covered in "Administration/maintenance" on page 46.

Apply and save changes

The web interface runs locally on the device, which means that the interface always maintains and displays the latest settings in the Digi device. On each screen, the **Apply** button is used to save any changes to the configuration settings to the Digi device.

Cancel changes

To cancel changes to configuration settings, click the **Refresh** or **Reload** button on the web browser. This causes the browser to reload the page. Any changes made since the last time the **Apply** button was clicked are reset to their original values.

Restore the Digi device to factory defaults

The device configuration can be reset to factory defaults as needed during the configuration process. See "Factory reset" on page 21. Note that any network configuration settings will have to be reset after the restore operation is complete.

Configuration settings pages

Ethernet Network

The **Ethernet Network** settings display the current IP address and DHCP settings for Ethernet network communications. The IP address can be changed, either by obtaining a new one through DHCP or by entering a static IP address.

The **Domain Name Service Configuration** show the two Domain Name System servers to be used as static servers when dynamic mechanisms do not supply enough DNS servers

The **iDigi Service Configuration** has one setting, **Server Address**. This setting is the address of the iDigi server used to manage the ConnectPort X2e ZB. The **Configuration > iDigi Connectivity** page enables or disables the connection to the specified iDigi server.

Wireless Network

The **Wireless Network** settings display the current IP address and DHCP settings for Wi-Fi network communications. The IP address can be changed, either by obtaining a new one through DHCP or by entering a static IP address.

Wireless Wizard

The Wireless Wizard link launches a wizard that is used to “teach” a Wi-Fi ConnectPort X2e ZB device the wireless parameters needed to further configure wireless settings and operation.The wizard provides a place to copy in the information required for the to connect to your local WiFi network. In this instance, the ConnectPort X2e ZB is a Wi-Fi “client” connecting to an existing access point. The information you are required to enter in this wizard should be obtained from the network administrator for the wireless network the ConnectPort X2e ZB will use. To complete the wizard:

1 Enter the SSID, or choose an SSID from the list of values, if supplied. This selection is required.

2 Enter the type of network security used on the wireless network.

3 Depending on the network security type, additional parameters can be entered.

4 Messages are displayed while the parameters are being processed and when the wizard is complete.

For security reasons, the wireless configuration resulting from running this wizard cannot be viewed, only modified by re-running the wizard.

Python: configure automatic starting and exit conditions for Python programs

The **Python Autostart Configuration** page configures the automatic starting of Python programs that are loaded on the ConnectPort X2e ZB device. See "File management" on page 48 for details on loading files. These programs are executed through the specified “python” commands.

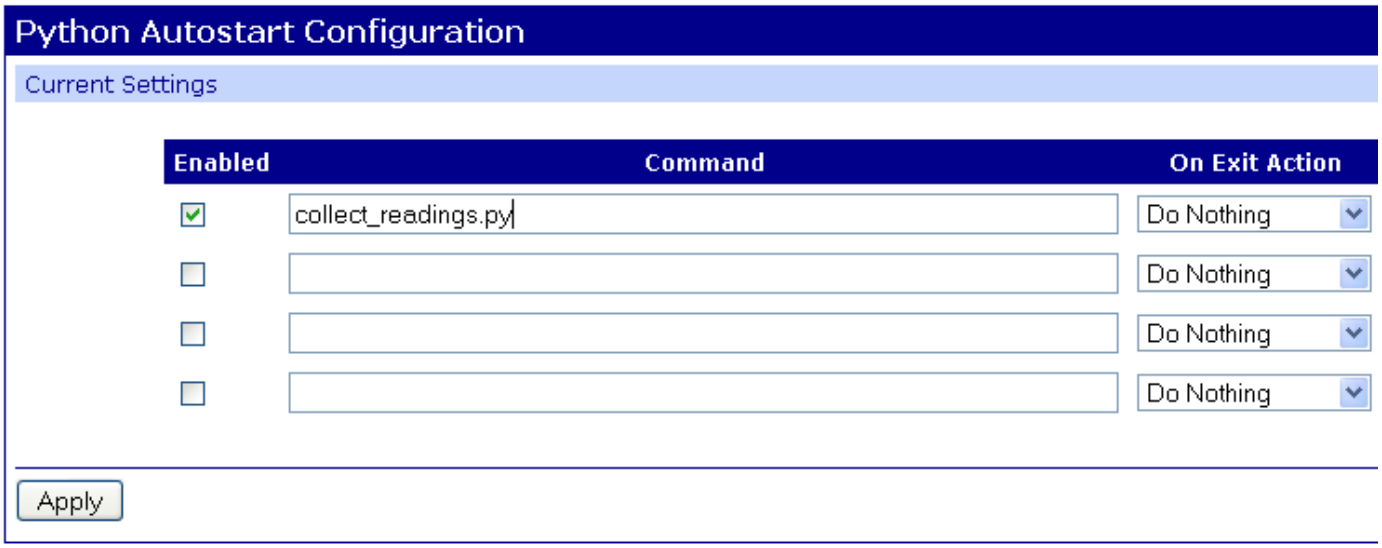
Select the **Enabled** checkbox to set the program to be automatically started at system startup.

In the **Command** field, enter the name of the Python file to be started, and any program arguments.

The **On Exit Action** setting controls the action to be performed in the event the program exits.

- **Do Nothing**: Continue device operation without doing anything about the program.
- **Reboot**: Reboot the device.
- **Restart the Script**: Restart the Python program.

For example:



Network Services

The **Network Services** page enables or disables common network services that are available on the ConnectPort X2e ZB, and configures the network port on which the service is listening. Disabling services may be done for security purposes. That is, certain services can be disabled so the device runs only those services specifically needed and to improve device security.

The only configurable network services on the ConnectPort X2e ZB are HTTP, HTTPS, and SSH. Following is more information about each service and its default network port number. It is usually best to use the default network port numbers for these services because they are well known by most applications.

Service	Services provided	Default network port number
HyperText Transfer Protocol (HTTP), also known as Web Server	Access to web pages for configuration that can be secured by requiring a user login. HTTP and HTTPS, below, are also referred to as Web Server or Secure Web Server. These services control the use of the web interface. If HTTP and HTTPS are disabled, device users cannot use the web interface to configure, monitor, and administer the device.	80
HyperText Transfer Protocol over Secure Socket Layer (HTTPS), also known as Secure Web Server	Access to web pages for configuration that can be secured by requiring a user login with encryption for greater security.	443
Secure Shell Server (SSH)	Allows users secure access to log in to the Digi device and access the command-line interface.	22

Time

The **Time Configuration** page configures the various Network Time Protocol (NTP) servers that the ConnectPort X2e ZB uses to obtain current date and time, and the time zone for the NTP server. The default settings for this feature are likely to be sufficient for many users.

iDigi Connectivity

The **iDigi Connectivity** page enables/disables the connection to the iDigi server used for managing the ConnectPort X2e ZB device, configures the iDigi server, and configures the proxy server, if used.

- **iDigi Connectivity**: Enables or disables the connection to the iDigi server.
- **iDigi Server Name/ iDigi Server Port:** Obtain this value from your iDigi server administrator. Typically this administrator is Digi. If your enterprise has its own iDigi server rather than one provided by Digi, enter your enterprise-specific iDigi server information here.
- **Proxy Server Name/port**: These values configure which system will provide network access, and which port number will forward network connection attempts to the iDigi server. Obtain these values from your iDigi server administrator.

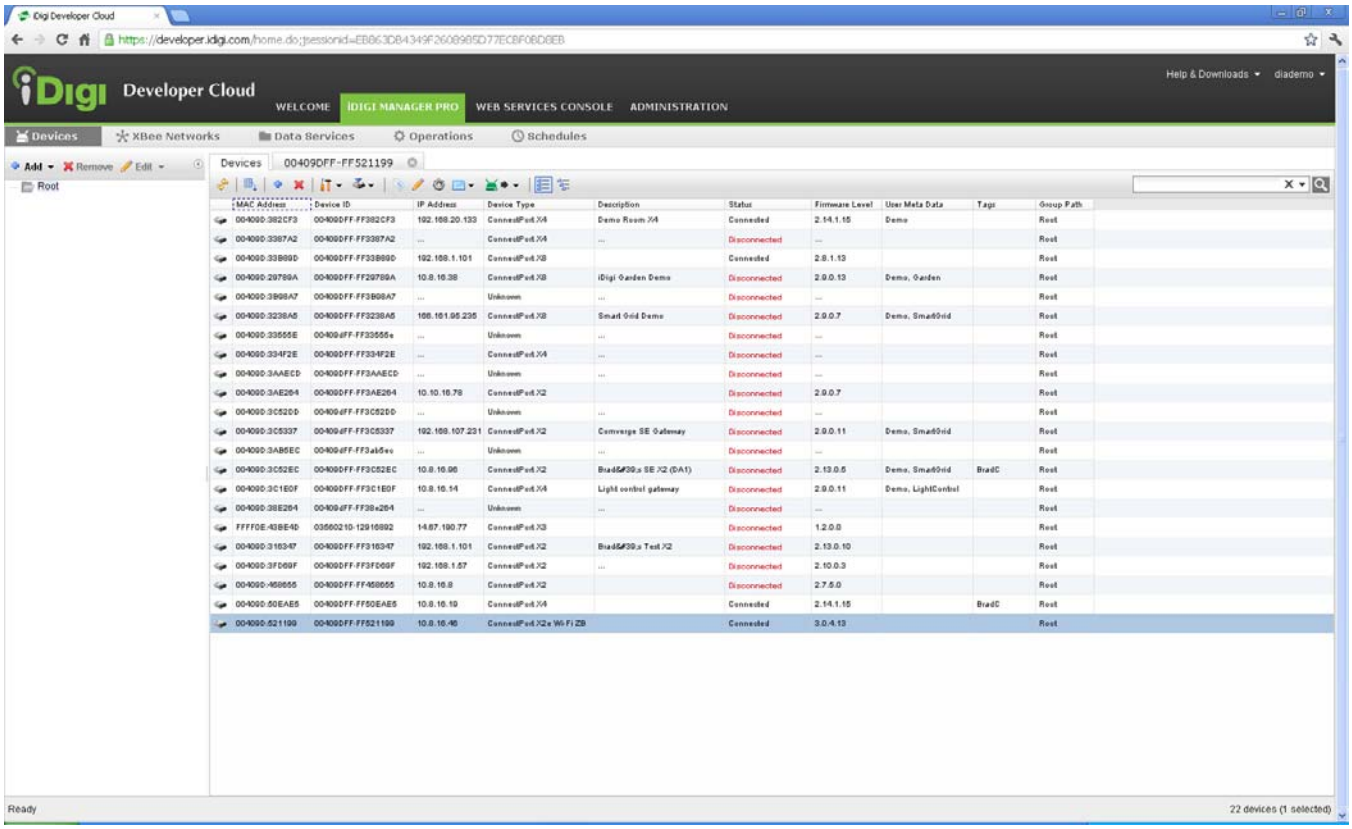
XBee Network

The ConnectPort X2e ZB provides a gateway between an Internet Protocol (IP) network and a network of various wireless devices containing XBee RF modules. Typically, these XBee devices are small sensors and controllers.

- **Current Devices:** Displays a view of XBee network devices, including the ConnectPort X2e ZB gateway and any nodes that have been discovered by the XBee module in the ConnectPort X2e ZB gateway. Each node displayed provides this information:
- **Node ID**: A descriptive, user-friendly name for the device. The Node ID is a 20-byte printable ASCII string that allows for referencing devices by names rather than their physical addresses.
- **Network**: A 16-bit network address for the device.
- **Extended address**: A static 64-bit address for the device that is a guaranteed to be unique.
- **Node type**: The node type for the device, or role it plays in the XBee network: **coordinator** or **router.**
- **Product type**: A description of the product type for the node, such as “X2e Gateway.”
- **Apply**: Applies the changes to the Node ID for the node.
- **Refresh**: refreshes the list of nodes that have joined the XBee network. (Note that the discovery operation may take a few seconds.) Refresh lists only nodes that respond to the discovery operation. Previous nodes are cleared from the list. Sleeping or unresponsive nodes may not be listed immediately, but may later appear in the list.

Configuration from iDigi Manager Pro

The *iDigi User's Guide* describes how to set up an iDigi account and add devices to the iDigi Manager Pro device list. Once your ConnectPort X2e ZB is added to the device list, you can view the device in the device list and access its configuration settings.



Basic configuration settings

In the iDigi Manager Pro interface, basic configuration settings for the ConnectPort X2e ZB are accessed by right-clicking the device in the device list and selecting **Properties**. The basic configuration settings located in this menu are:

- Ethernet settings
- Wi-Fi settings
- Domain Name Server (DNS)
- Python

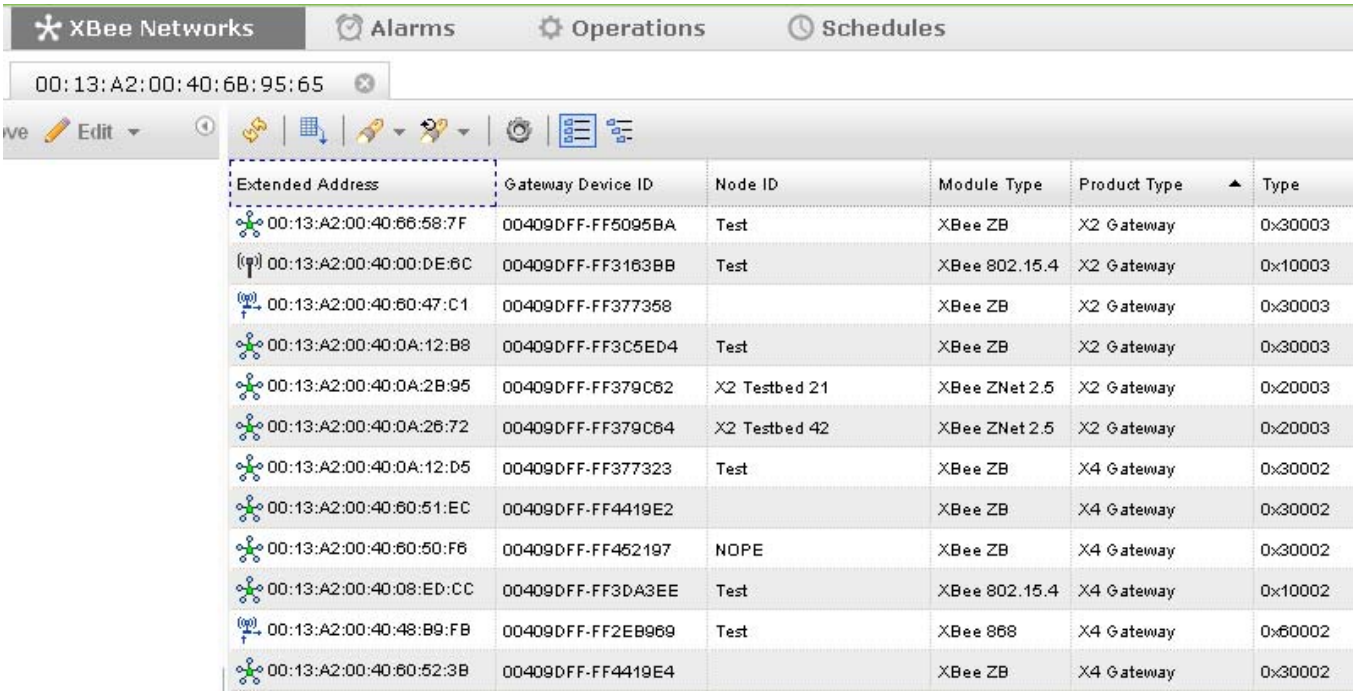
There are several other operations on the context menu, such as **File Management**, **System Information**, and **Connection History**. These operations are important for administrative and troubleshooting tasks and are discussed elsewhere in this guide.

Advanced configuration settings

Advanced configuration settings for the ConnectPort X2e ZB are accessed by right-clicking the device in the device list and selecting **Properties > Advanced Configuration**. The settings available in this menu vary by model.

XBee Networks page

Clicking the **XBee Networks** menu within the iDigi Manager Pro tab opens the **XBee Networks** page. This page allows you to manage all the XBee Nodes in your inventory.



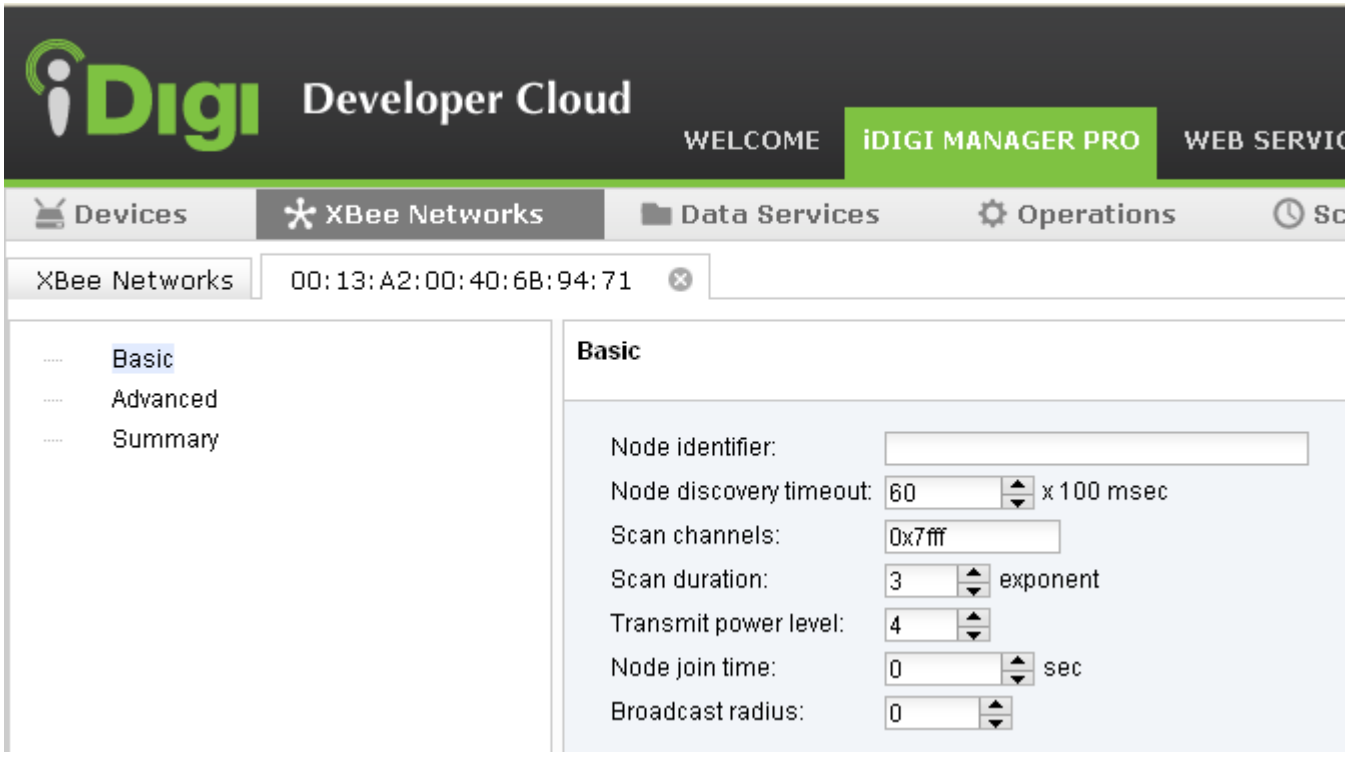
Extended Address	Gateway Device ID	Node ID	Module Type	Product Type	Type
00:13:A2:00:40:66:58:7F	00409DFF-FF5095BA	Test	XBee ZB	X2 Gateway	0x30003
00:13:A2:00:40:00:DE:6C	00409DFF-FF3163BB	Test	XBee 802.15.4	X2 Gateway	0x10003
00:13:A2:00:40:60:47:C1	00409DFF-FF377358		XBee ZB	X2 Gateway	0x30003
00:13:A2:00:40:0A:12:B8	00409DFF-FF3C5ED4	Test	XBee ZB	X2 Gateway	0x30003
00:13:A2:00:40:0A:2B:95	00409DFF-FF379C62	X2 Testbed 21	XBee ZNet 2.5	X2 Gateway	0x20003
00:13:A2:00:40:0A:26:72	00409DFF-FF379C64	X2 Testbed 42	XBee ZNet 2.5	X2 Gateway	0x20003
00:13:A2:00:40:0A:12:D5	00409DFF-FF377323	Test	XBee ZB	X4 Gateway	0x30002
00:13:A2:00:40:60:51:EC	00409DFF-FF4419E2		XBee ZB	X4 Gateway	0x30002
00:13:A2:00:40:60:50:F6	00409DFF-FF452197	NOPE	XBee ZB	X4 Gateway	0x30002
00:13:A2:00:40:08:ED:CC	00409DFF-FF3DA3EE	Test	XBee 802.15.4	X4 Gateway	0x10002
00:13:A2:00:40:48:B9:FB	00409DFF-FF2EB969	Test	XBee 868	X4 Gateway	0x60002
00:13:A2:00:40:60:52:3B	00409DFF-FF4419E4		XBee ZB	X4 Gateway	0x30002

The *iDigi User's Guide* provides describes the settings and operations that can be performed from the XBee Networks pages and menus.

XBee RF module settings

The configuration information for an XBee node is displayed by opening its **Properties** page. To do this, either double-click on the node in the XBee Networks page, or right-click the node and select **Properties** or click the **Properties** toolbar button. The **Basic Settings** control basic operation of the XBee module in an XBee network. **Advanced radio settings** control behavior of the XBee module at a more detailed level. The **Summary** view provides an overview of current XBee RF module operational and state information.

If you select multiple devices, multiple **Properties** pages will be open corresponding to each device that was selected.



Administration/maintenance

CHAPTER 4

This chapter describes the administrative and maintenance tasks that need be performed for the ConnectPort X2e ZB periodically. It covers common administrative tasks and how to perform them through iDigi Manager Pro and the web interface.

Common administrative tasks

For the ConnectPort X2e ZB, common administrative tasks include:

- Firmware updates
- File management
- Rebooting the device

Firmware updates

There are two types of firmware updates for the ConnectPort X2e ZB:

- Firmware updates for the gateway operating system
- Firmware updates for the for the XBee RF module on the gateway and nodes

Gateway operating system firmware updates

Gateway operating system firmware updates for the ConnectPort X2e ZB can be downloaded from the Digi Support site. They can then be loaded through the web interface or iDigi Device Manager Pro.

The gateway operating system can only be upgraded, not downgraded, through the web interface or iDigi Manager Pro. The firmware image contains a certificate that is verified before an upgrade can be accepted by the gateway.

XBee RF module firmware updates

Updating the firmware for XBee RF modules involves two firmware update scenarios:

- Update firmware for the XBee module on the gateway
- Update firmware on XBee modules on nodes elsewhere in the network. This type of update is known as an *over-the-air (OTA) firmware update*.

Download firmware update files from Digi

There is only one firmware file for the XBee RF module in ConnectPort X2e ZB products. It is named **XBP24-S2C_40*XX*.ebl**, where ***XX*** is the firmware version.

This firmware is located on the Digi Support site at:

Support->XBee/XBee-PRO ZB SMT (S2C)Modules->Firmware Updates

From this location, you will download the **.zip** file; the **.ebl** file is included in the **.zip** file.

Then, you can use the **Firmware Update** function from the web interface or iDigi Manager Pro to update the firmware.

File management

A file management feature is provided to manage custom applications, their associated data files, and other files. It is also possible to push firmware update files to the device filesystem, and to pull log files from the device filesystem. The device filesystem is described on page 67.

Reboot device

The device can be rebooted as needed from the web interface or iDigi Manager Pro.

Administration from the Web interface

Firmware Update

The **Firmware Update** page updates the gateway operating system firmware. (Firmware updates for the XBee RF module on the gateway or nodes are done on the **XBee Firmware Update** page.)

- **Volume Information**: Displays the current directory and free space remaining.
- **Firmware Information**: Displays the current firmware version running on the device, and its state: Running, Updating, Recent update failed.
- **Upload to Current Directory:** Uploads the firmware file to the current directory. Use the **Browse** button to choose the file on your PC that is the firmware update image.
- Click **Update Firmware** to pull the file from the PC into the device, then attempt a firmware update with the selected image.

XBee Firmware Update

The **XBee Firmware Update** page updates the firmware for the XBee RF module on the gateway.

- **Volume Information**: Displays the current directory and free space remaining.
- **Firmware Information**: Displays the current firmware version running on the device, and its state.
- **Upload to Current Directory:** Uploads the firmware file to the current directory. Use the **Browse** button to choose the file on your PC that is the firmware update image. Then use the **Upload** file button to upload the XBee firmware file to the **/userfs/WEB/mesh** directory in ConnectPort X2e ZB filesystem.
- Select the appropriate firmware file from the **File List**, and then click the **Update XBee Firmware** button to attempt a firmware update with the selected image.

XBee Status

The **XBee Status** page displays status information for the XBee RF module on the gateway. The parameters displayed vary based on the capabilities supported by XBee RF module; common parameters include the PAN ID, firmware and hardware versions, and the device type identifier. See the *XBee/XBee-PRO ZB SMT RF Modules Product Manual* for parameter descriptions.

File Management

The **File Management** page loads files onto the ConnectPort X2e ZB and displays current information about loaded files. For a description of the filesystem layout, see page 67.

- **Volume Information**: Displays the current directory for loading files and free space remaining.

  Note: Digi recommends using no more than 20 MB for custom applications, as the ConnectPort X2e ZB requires a portion of the same space to be capable of managing persistent system logs and firmware updates.

- **Upload to Current Directory**: Uploads files to the current directory. Use the **Browse** button to find a file on your PC to be uploaded to the ConnectPort X2e ZB. Click **Update file** to begin the file upload process.

- **File List:** A listing of the **"**current directory**"** (as noted in the **Volume Information** area) on the device.
  - **Open**: Opens a directory after it is selected in the file list. The **"**current directory**"** will change and the list will be updated.
  - **Make Directory**: Creates a new, empty directory in the **"**current directory.**"**
  - **Save As**: Downloads a regular file from the filesystem to your local PC.
  - **Remove**: Deletes files or, for directories, empty directories only.

- **Refresh button**: Reloads all the information on the page.

System Log

The **System Message Log File Browser** is a low-level debugging and diagnostic tool that allows viewing entries in a system log file. The default log file that can be viewed from this browser is **digi.log**. This file is the primary message log for informational notices. This log is persistent and can have data from multiple boots in it. But because it has a maximum size, it is not guaranteed to contain all of the device history.

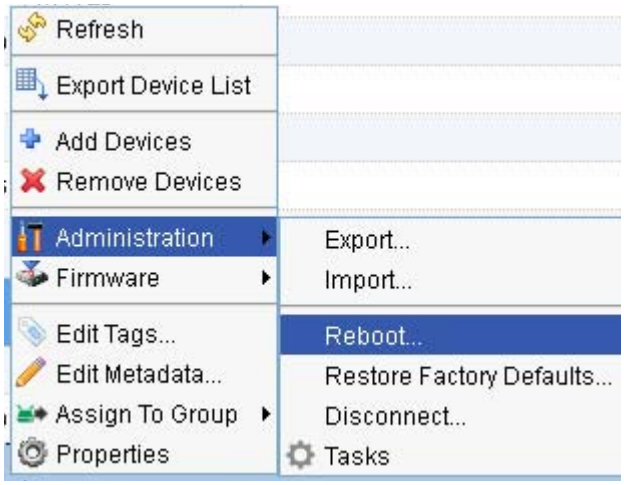
The **Message Filters** allow Digi Technical Support working with customers on troubleshooting issues to more quickly hone in on areas of interest in the log file as needed. The names of the filters correspond to features in the ConnectPort X2e ZB and are Digi-internal.

Reboot

To reboot the ConnectPort X2e ZB, click the **Reboot** menu option. Wait approximately 1 minute for the reboot to complete.

Administration from iDigi Manager Pro

In iDigi Manager Pro, administration tasks are organized in the right-click menu for the ConnectPort X2e ZB, under **Administration** and **Firmware**. The File Management task is launched from a link in the device **Properties** page.



Restore Factory Defaults

Administration > Restore Factory Defaults restores the factory default configuration to the selected device or devices. The only supported choice is to reset all the configuration settings.

Reboot

Administration > Reboot reboots the selected device or devices.

Disconnect

Administration > Disconnect disconnects the device from iDigi. If the device is configured to automatically reconnect to iDigi, it will re-establish the connection to iDigi after its reconnection timeout period has expired.

Firmware

The Firmware options perform several types of firmware updates.

Update Firmware: Gateway operating system firmware updates

Firmware > Update Firmware updates the gateway operating system firmware from a specified firmware file. In the dialog, select a firmware file to upload to the selected device or devices. Once the firmware file is downloaded to the device, the device will be rebooted automatically.

Update Gateway XBee Radio Firmware

Firmware > Update Gateway XBee Radio Firmware updates the firmware on the XBee RF module on the ConnectPort X2e ZB. In the dialog, select an XBee radio firmware file to upload to the selected device or devices. Once the XBee radio firmware file is downloaded to the device, the device will be rebooted automatically.

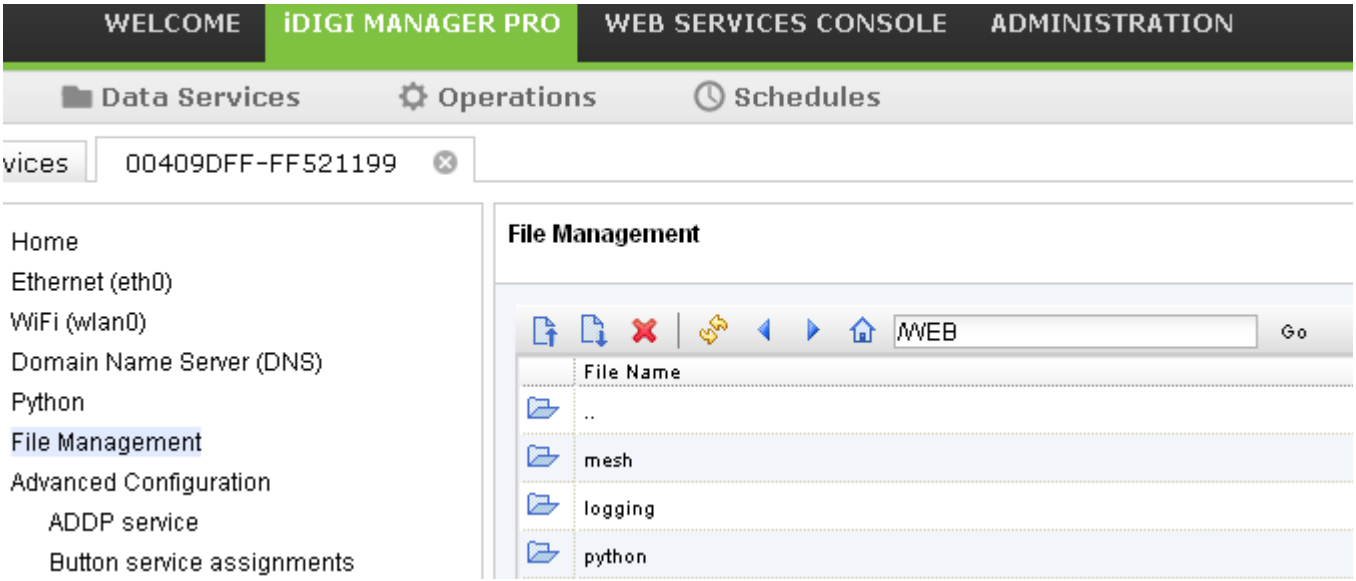
Update XBee Node Firmware

Firmware > Update XBee Node Firmware updates the XBee firmware for the selected remote node. Once the XBee radio firmware file is downloaded to the remote node, the remote node will be rebooted automatically.

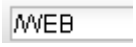
File Management

The File Management task is launched from the **Properties** page for ConnectPort X2e ZB, by clicking **File Management**. This page loads files, such as custom application files, onto the ConnectPort X2e ZB and displays current information about loaded files. This page can also be used to download system log files from the ConnectPort X2e ZB to view for troubleshooting or other purposes. These log files are in the **/WEB/logging** folder. For a description of the ConnectPort filesystem layout, see page 67.

Note: Digi recommends using no more than 20 MB for custom applications, as the ConnectPort X2e ZB requires a portion of the same space to be capable of managing persistent system logs and firmware updates.



The File Management toolbar provides quick access to file management tasks.

Button/ Field	Name	Description
	Upload file	Opens a dialog for uploading files to the current folder (directory).
	Download file	Downloads the selected file to a PC. You can choose to open the downloaded file with a specfied tool or save it.
	Delete	Deletes the selected items.
	Refresh	Refreshes the list of folders and files displayed.
	Back	Move to the previous folder in folder history.
	Forward	Move to the next folder in folder history.
	Home	Returns to the root folder.
	Current or destination folder	An editable field that displays the current folder, and which can be used to enter a different destination directory.
	Go	Goes to the directory specfied in **Current or desination folder** field..

System log

There is no browser interface for system logs in the iDigi interface. Instead, to view logs, use the File Management page to navigate to the directory **WEB/logging** in the ConnectPort X2e ZB filesystem, and access the log files **digi.log**, **xbee.log**, and **python.log** using the **Download file button**. These files are read-only.

Programming

C H A P T E R 5

The ConnectPort X2e ZB offers a variety of interfaces that produce and/or consume data. Developing software programs for ConnectPort X2e ZB products allows Digi customers to provide custom logic to control the flow information to and from these interfaces. This chapter introduces the Digi programming tools and resources available to you, and, using annotated example programs, introduces programming elements and operations available for the ConnectPort X2e ZB.

Programming resources

Several tools and resources are available for developing software programs for the ConnectPort X2e ZB.

Python

Python is a dynamic, object-oriented language for developing software applications, from simple programs to complex embedded applications. The ConnectPort X2e ZB runs the standard Python 2.7.1 distribution. This Python distribution has a more complete library set than the versions supported in predecessor gateway products, and integrates well with standard Python.

Digi ESP for Python

Digi ESP for Python is an Integrated Development Environment (IDE) that allows for easy development of applications in the Python programming language to be loaded, run, tested, and debugged on the device. This development environment includes a Getting Started Guide that guides you in creating your first simple application for your ConnectPort X2e ZB, plus extensive documentation, tutorials, and code samples to help you get started developing custom applications for your product. Digi ESP for Python is built on Eclipse, an open, extensible IDE, and the Python Development Environment (Pydev) plug-in.

Unlike proprietary embedded development platforms, the integration of the universal Python programming language allows for a truly open standard for complete control of connections to devices, the manipulation of data, and event-based actions.

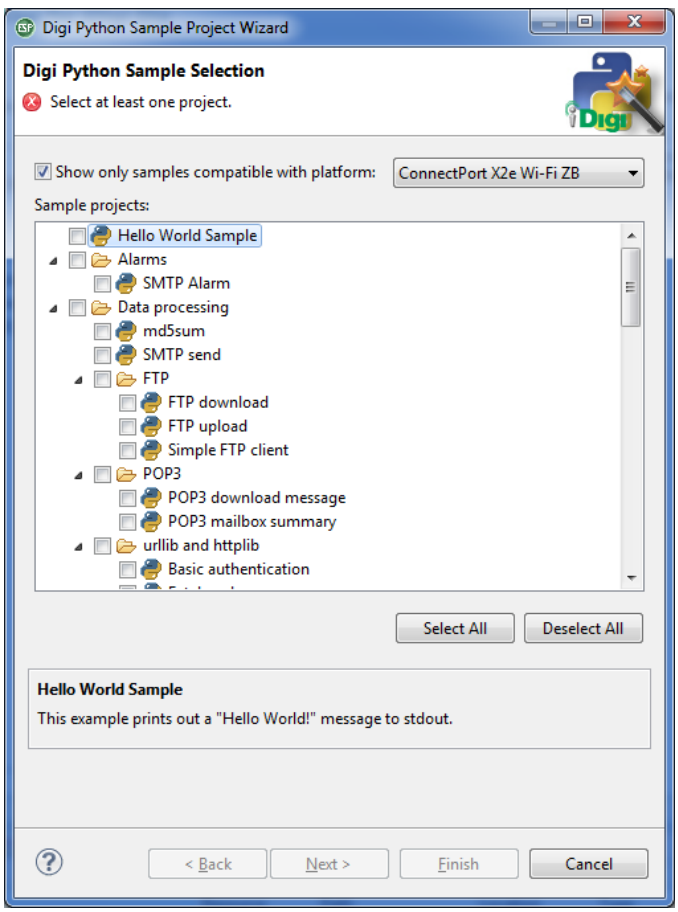
For example, applications can be created to:

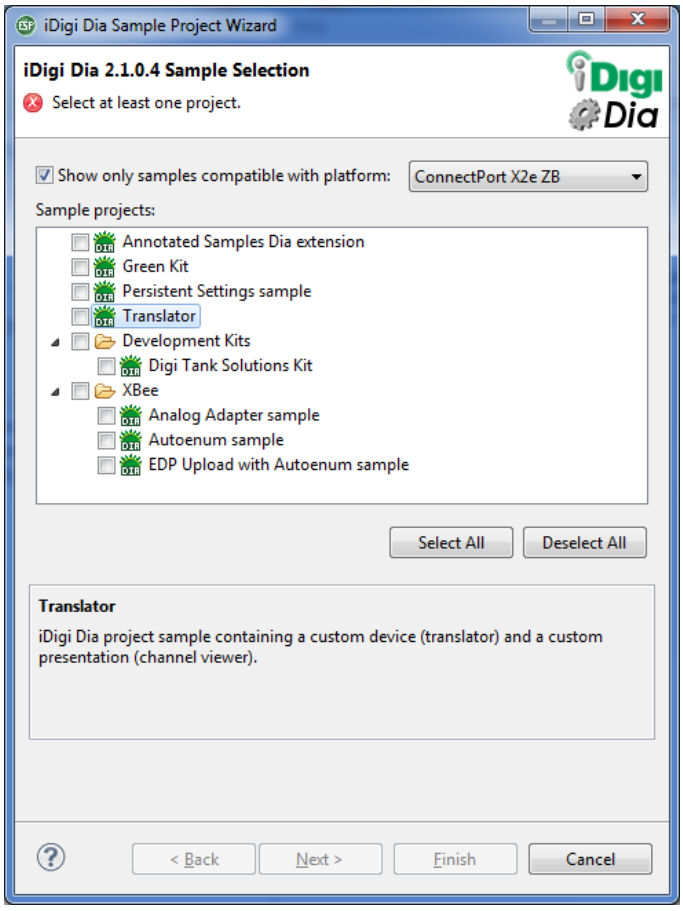
- Aggregate data on the gateway throughout the day, then upload to iDigi once a day.
- Develop a driver for a custom protocol to talk to device connected to the gateway, such as devices in an XBee network.
- Translate data, for example, from raw serial into an XML format that can be consumed by other systems, such as iDigi.

Program samples in Digi ESP

Digi ESP for Python provides samples to use as a base for programming the ConnectPort X2e ZB, including specific examples for exercising specific interfaces available on the ConnectPort X2e ZB. These program samples can be accessed from the Python or iDigi Dia samples wizard.

1. Go to **File > New > Digi Python Application Sample Project or File > New > iDigi Dia Sample Project.** The samples wizard, Python or iDigi Dia, will be opened.
2. Check the option **Show only samples compatible with platform.** In the combo box, select **ConnectPort X2e ZB** or **ConnectPort X2e Wi-Fi ZB**.
3. The samples in the list will be filtered, and only the ones compatible with the selected sample will be displayed.; for example:





iDigi Dia software

The Digi ESP development environment includes iDigi Device Integration Application (iDigi Dia) software. This software simplifies connecting devices (sensors, PLCs, etc.) to communication gateways. It includes a comprehensive library of plug-ins that work out-of-the-box with common device types and can be extended to include new devices. Its unique architecture allows the user to add most devices in under a day. iDigi Dia is a tested architecture that provides the core functions of remote device data acquisition, control, and presentation between devices and information platforms. It collects data from any device that can communicate with a Digi gateway, and is supported over any gateway physical interface. iDigi Dia presents this data to upstream applications in fully customizable formats, significantly reducing a customer's time to market. Written in the Python® programming language for use on Digi devices, iDigi Dia can be executed on a PC for prototyping purposes when a suitable Python interpreter is installed. iDigi Dia is targeted for applications that need to gather samples of data from a set of devices, such as ZigBee® sensors, wired industrial equipment, GPS devices, etc. It is an integral component of the iDigi platform, which customers can deploy with iDigi Dia software to build flexible, robust solutions with unprecedented speed.

Programming calls through Server Command Interface (SCI) and Remote Command Interface (RCI)

The iDigi Web Services Server Command Interface (SCI) and the Remote Command Interface (RCI) can be used as an alternative means of getting settings and state data from the device. See the SCI (Server Command Interface) chapter of the *iDigi Web Services Programming Guide* and the *Remote Command Interface (RCI) Specification.*

Digi Developer Community Wiki

The Digi Developer Community Wiki provides references for developing solutions using Digi's communications portfolio, software and services, including Python, iDigi Platform, iDigi Dia, and more.The Wiki includes how-to's, example code, and M2M information to speed application development. Digi encourages an active developer community and welcomes your contributions.

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

Within this Wiki are pages for ConnectPort X2e ZB-specific programming information, beginning here:

http://www.digi.com/wiki/developer/index.php/ConnectPort_X2e

Digi Python Programmer’s Guide

The *Digi Python Programmer’s Guide* introduces the Python programming language by showing how to create and run a simple Python program. It reviews Python modules, particularly those with Digi-specific behavior, and describes how to load and run Python programs onto Digi devices, and run sample Python programs. This guide is on the Digi Python Wiki; in the **Start Here** section, click the link titled **Digi Python Programmer's Guide**

http://www.digi.com/wiki/developer/index.php/Digi_Python_Programmer%27s_Guide

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/listforums?category=25

Digi-specific Python modules for programming

Several Digi-provided built-in modules apply to functionality in the ConnectPort X2e ZB. To learn more about these built-in modules, go to:

http://www.digi.com/wiki/developer/index.php/ConnectPort_X2e

From that page, navigate to the information pages on these modules:

- **digihw**: Provides an interface to local hardware.
- **idigimsg**: This is an internal module with functions used by iDigi Dia and Smart Energy Framework.
- **rci**: Provides a mechanism for processing arbitrary RCI request strings, as well as a means to set up callbacks to allow Python scripts to respond to remote requests made via specialized RCI commands.
- **uptime**: Allows access to the Linux **gettime** function for various clocks.
- **watchdog**: A safeguard that an application can use for critical operations, in which, if the application threads do not service their watchdog within the promised interval, the entire system reboots.
- **xbee**: A collection of utility methods for programming the XBee RF module on the ConnectPort X2e ZB device. It also includes the Python XBee socket interface support.

Sample programs

These simple annotated samples introduce several programmable features for the ConnectPort X2e ZB, including button handling, LED control, watchdog, RCI callback, and XBee functions.

Button handling

The following sample program demonstrates functions for handling the button on the ConnectPort X2e ZB.

```
import select

fd=open('/var/run/reset_button')                                    (1)
p=select.poll()
p.register(fd, select.POLLPRI)                                      (2)
fd.read()                                                           (3)


while True:
      p.poll()                                                      (4)
      fd.seek(0)                                                    (5)
      val = int(fd.read()[0])                                       (6)

      if val:                                                       (7)
            print "Button pressed!"
      else:
            print "Button released!"
```

Program notes

1 The reset button is exposed as a Linux file. It can be read to determine the state of the button, and it is possible to block waiting for the button state to change.

2 To block waiting for the button, the standard Python **select** module is used. This line, and the line above, demonstrate how to create a polling object that can wait for button state changes.

3 Read the current value of the button, but forget it. This is done to "clear" the button and prepare to wait for its state changes.

4 Rather than reading the button in a loop, the system waits for button state changes using the polling object created earlier.

5 To read the current value, we first "rewind" to the beginning of the "file."

6 **fd.read()** gets pending data from the button file. **fd.read()[0]** returns just the first character of that data. **int(fd.read()[0])** makes explicit the fact that we expect the character we read to be an integer.

7 If we read a non-zero value, the button is currently pressed.

LED control

Controlling the LEDs on the ConnectPort X2e ZB is handled through the **user_led_set** function. This sample program controls the Network LED.

Note that if native features are still assigned to the LED, those behaviors will mix with the behaviors in the sample program.

```
import digihw                                                    (1)
import time

while True:
      digihw.user_led_set(True,1)                                (2)
      time.sleep(1.0)
      digihw.user_led_set(False,1)                               (3)

      digihw.user_led_set(True, 2)                               (4)
      time.sleep(1.0)
      digihw.user_led_set(False,2)                               (5)
```

Program notes

1 The **digihw** module includes the **user_led_set** function needed by the program. **user_led_set** has two are two parameters: *value* and *led*. The user controlled LED is made to match the logic state of the "*value*" parameter. A value of "True" turns on the LED, and a value of "False" turns it off. The "led" parameter indicates which user LED to blink, with LED 1 being the default.

2 Turn on the "yellow" Network LED.

3 Turn off the "yellow" Network LED.

4 Turn on the "green" Network LED.

5 Turn off the "green" Network LED.

Watchdog

The watchdog feature, provided through the **watchdog** module, exists as a safeguard. If there are critical operations that "must" happen periodically, or else the system will be irretrievably broken, an application can request that a "watchdog" be established. If the application threads do not service their watchdog within the promised interval, the entire system reboots. These software watchdogs can have their intervals changed, if necessary, and can be deleted. Clearly, use of such a software watchdog exists as a measure of last resort. Appropriate error detection and handling with Python scripts is certainly recommended.

The following sample program demonstrates the watchdog feature.

```
import watchdog                                                    (1)
import time

w=watchdog.Watchdog('test',20)                                     (2)
for x in xrange(1,6):                                              (3)
  print "Step ", x                                                 (4)
  time.sleep(10.0)                                                 (5)
  w.heartbeat()                                                    (6)
print "Step just before the end..."                                (7)
time.sleep(60.0)                                                   (8)
print "Step after the end."                                        (9)
```

Program notes

1 The **watchdog** module includes the **Watchdog** class needed by the program.

2 Create a watchdog object named "test" that will expire in 20 seconds.

3 Loop five times (1-5).

4 Indicate our iteration...

5 ... sleeping less than the timeout on each iteration, but more time than the timeout in total.

6 Reset the watchdog timer to 20 seconds each iteration, allowing all of the loops to complete.

7 Indicate that small loops are complete.

8 Sleep for an interval much longer than the timeout.

9 This **print** statement should never be executed, as the system should have rebooted due to the watchdog timeout expiration.

RCI callback

An RCI callback involves two types of actions, demonstrated in the following programs:

- Making RCI requests from Python applications
- Extending RCI to allow iDigi to make requests of Python applications. This is known as an *RCI callback*.

Make an RCI request-example program

```
import rci                                                          (1)

request_string="""                                                  (2)
<rci_request version="1.1">
  <query_state>
   <interface_info name="eth0">
    <ip/>
   </interface_info>
  </query_state>
</rci_request>
"""

print rci.process_request(request_string)                           (3)
```

Program notes

1 The **rci** module includes the **process_request** function needed by the program.

2 A string representing the RCI request is needed. The sample uses the Python multi-line string syntax to make it clearer that the XML represents a request for the current IP address of the Ethernet interface. Combining the lines into a single string on one line would work in the same way.

3 The RCI XML is submitted for parsing, and the resulting string is returned. In this sample, the result is simply printed.

Make an RCI request-example program

Following is a simple RCI callback example:

```
import rci                                                          (1)

def cb(req):                                                        (2)
      print "Received request: " + req                              (3)

r=rci.RciCallback()                                                 (4)
r.register_callback('test', cb)                                     (5)

rci.process_request('<rci_request version="1.1"><do_command
  target="test"><customxml/></do_command></rci_request>')           (6)
```

Program notes

1 The **rci** module includes the **RciCallback** class needed by the program.

2 Create a function to be called whenever a remote entity wishes to communicate with this script.

3 This simple function will simply demonstrate that it received a request that could be parsed and handled however the application saw fit.

4 Create a callback object.

5 Assign the target "**test**" to the new callback object. If a remote entity issues a **"do_command"** with the target **"test"**, the supplied callback function will be called.

6 This is simply an example that causes the callback to be called. This example could also have been a remote SCI query through iDigi.

XBee functions

For a description of the **xbee** module and program samples, see this Wiki page:

http://www.digi.com/wiki/developer/index.php/Module:zigbee

The ConnectPort X2e ZB filesystem

The ConnectPort X2e ZB has a Linux-based filesystem. This section gives an overview of the key directories of the filesystem and common operations performed on directories and files.

Differences between Windows and Linux filesystems

If you are transitioning from a Windows-based filesystem to a Linux filesystem, the following article covers some of the key differences between Linux and Windows filesystems.

http://freeos.com/articles/3102

Important directories

The files on a Linux system are arranged in what is called a hierarchical directory structure. This means that they are organized in a tree-like pattern of directories (called folders in other systems), which may contain files and other directories. The first directory in the file system is called the **root** directory. The **root** directory contains files and subdirectories, which contain more files and subdirectories and so on and so on. One important difference between the legacy operating system and Unix-like operating systems such as Linux is that Linux does not employ the concept of drive letters. While drive letters split the file system into a series of different trees (one for each drive), Linux always has a single tree. Different storage devices may contain different branches of the tree, but there is always a single tree.

The frontslash **/** is used as a directory separator, instead of the backslash \ used in Windows and DOS.

The **/WEB/python/** directory is for user-specific files, such as custom Python applications. Subdirectories can be created in this area for the customer's applications. This area is read-write.

The **/WEB/logging** directory contains system log files, including **digi.log**, **xbee.log**, and **python.log**. These files are read-only.

Access/browse the filesystem from device interfaces

There are three aspects or ways in which you will interact with the filesystem resident on the ConnectPort X2e ZB: through the web interface, and through the command line interface, and through iDigi.

Common operations for directories and files

The ConnectPort X2e ZB supports standard Linux shell file operations for managing directories and files. See the command shell reference document listed on page 68 for details on these operations.

The Linux command shell (command-line interface)

A *shell* is a program that takes commands from the keyboard and gives them to the operating system to perform. On the ConnectPort X2e ZB, a a program called **ash** acts as the shell program. This shell can be useful in several instances, particularly in managing files, executing Python programs, and programming and executing commands the XBee RF module in the gateway.

Username and password for the Linux command shell

Access to the ConnectPort X2e ZB is at the user level

The username is **python**.

The password is **dbps**.

Connect and log on to the device

To connect and log on to the device through the command-line interface:

1. Open a command prompt or terminal window, such as one provided by PuTTY available for downloading from http://www.putty.org/.
2. From **Windows**, enter:

   ```
   putty SSH
   ```

   When prompted, enter the username: **python** and the password: **dbps**

   OR

   From **Linux**, enter:

   ```
   $ ssh python@my_ip_address
   ```

   Where *my_ip_address* is the IP address of the ConnectPort X2e ZB. When prompted, enter the password: **dbps**

Command shell reference documentation

For more information on the ash shell, see the following document. Note that the ash shell supported in the ConnectPort X2e ZB is similar, but not identical, to the ash shell described in this documentation.

http://linux.die.net/man/1/ash

Troubleshooting

CHAPTER 6

This chapter provides information on resources and processes available for troubleshooting your ConnectPort X2e ZB product.

System logs

The ConnectPort X2e ZB supports continuous logging to enable debugging. The log files are available for upload through the web interface or by pulling the files from the iDigi interface. They are stored in the Linux filesystem in the **/WEB/logging** directory. Log files in this directory include:

- **digi.log**
- **python.log**
- **xbee.log**

Troubleshooting chart

Symptom	Potential Cause	Resolution
Power LED is dark.	Power is not applied.	Ensure that the power supply is properly connected to a power source and the device.
	Hardware failure.	In the unlikely case of hardware failure, first physically remove power and reconnect. If the LEDs remain dark, the device should be returned for RMA.
XBee LED is dark.[1]	If device is configured as a coordinator, this may indicate a coordinator hardware failure	Return device to Digi for RMA.
Network LEDs are dark[1]	Hardware failure.	Return device to Digi for RMA.
Network LED is flashing yellow slower than once per second.[a]	No Ethernet link.	Attach an Ethernet cable.
	No Wi-Fi association.	Configure Wi-Fi network parameters. See page 29.
Network LED is flashing yellow faster than twice per second.	DHCP server not responding.	If no DHCP server is present on the network, use a tool with integrated Digi Device Discovery to assign a static IP address.
		Device may not have network connectivity to the DHCP server, despite having link to a hub or switch. Check with network administrator.

Table notes:

1 XBee and network LEDs remain dark during the initial boot of the device. All diagnosis related to these LEDs must be done after the device has been running for at least 60 seconds.

2 For devices with both Ethernet and Wi-Fi interfaces, the yellow network LED will communicate information about Ethernet if there is physical link detected on the Ethernet. Connectivity to iDigi over any interface (the green network LED) always overrides any yellow LED indication.

Symptom	Potential Cause	Resolution
Network LED alternates between yellow and green	Unable to resolve iDigi server name.	Verify DNS server settings with network administrator. See also page 72.
		Network failure between the device and the DNS server. Check with network administrator.
	Unable to contact iDigi server.	Verify iDigi server name in settings.
		Verify proxy settings with device and with network administrator.
		Network failure between the device and the iDigi server. Check with network administrator.
	Device unable to authenticate iDigi server.	Verify NTP server settings with network administrator. The device must have the correct time in order to verify the security certificate presented by the iDigi server. The supported means to synchronize time in the device is via NTP. See also page 72.
		Verify iDigi server name in settings.

a.

Cannot connect to DNS server to resolve the iDigi server address

The ConnectPort X2e ZB requires a DNS server to resolve the iDigi server IP address. Normally the device receives the DNS server information automatically from the DHCP server on your Internet router, but if your router does not provide the information, you may need to set it manually. When this error occurs, the system log (described on page 54) will display the following message:

```
Error resolving 'fully qualified domain name': No address associated
with nodename (EAI_NODATA)
```

The device usually gets the DNS server from the DHCP server. If a DNS server is not supplied from your DHCP server, usually because it is not configured to supply it, the ConnectPort X2e ZB will use the default Google Public DNS server (address **8.8.8.8**). If use of this default DNS server is not desired, a DNS server can be configured manually using the Web interface or RCI configuration interface.

It is most effective to fix the DHCP server to provide the correct networking parameters to the device. Otherwise, the device's network configuration can be set manually. To set the DNS server manually through the web interface, go to the **Configuration > Ethernet Network** page. See "Ethernet Network" on page 37. Contact your network administrator or ISP to get the DNS server information.

Cannot connect to NTP time server to get correct time

The ConnectPort X2e ZB requires access to an NTP server to set its date and time. Normally the device receives the date and time automatically from the iDigi server.

Contact your network administrator or ISP to find out if they have blocked access or have set up other NTP servers on their network. To change the NTP server through the web interface, go to the **Configuration > Time** page. See "Time" on page 39.

For NTP to operate correctly, UDP port 123 needs to be open.

Required open network ports and use of firewalls

When using a firewall to filter outbound traffic, the ConnectPort X2e ZB requires these network ports to be open for proper operation:

- UDP port 53, for DNS
- UDP port 123, for NTP
- TCP port 3199, for iDigi