AnywhereUSB® Plus

User Guide

Firmware version 21.2

Revision history—90002383

Revision	Date	Description
D	June 2020	Updated CLI command information for AnywhereUSB groups and clients.
E	July 2020	■ Highlighted information about changing the default password for the Hub. ■ Added troubleshooting topics: Invalid client certificate, Invalid client ID, Invalid Hub certificate, Red X icon next to a Hub in the AnywhereUSB Manager. ■ Added additional information to View Hub system information.
	August 2020	Release of Digi AnywhereUSB Plus firmware version 20.5: ■ Support for LDAP user authentication. ■ Firmware installation from the Digi firmware server. ■ Enhanced Digi Remote Manager support: • Support for remote proxy server for Digi Remote Manager. • Watchdog support for connection to Digi Remote Manager. • **Locally authenticate CLI** option added to Digi Remote Manager configuration to control whether a user is required to provide device-level authentication when accessing the console of the device through Digi Remote Manager. • Added a randomized two minute delay window for uploading health metrics to the Digi Remote Manager to avoid situations where multiple devices are uploading metrics at the same time. Additional updates: ■ Updated information about service and stand-alone modes. See Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone.

Revision	Date	Description
F	September 2020	Release of Digi AnywhereUSB Plus firmware version 20.8: ■ Support for NEMO/DMNR virtual private networks. ■ Support for Ethernet network bonding. ■ Support for VRRP+, an extension to the VRRP standard that uses network probing to monitor connections through VRRP-enabled devices. ■ Cloud service enhancements: • Reduced data usage for reporting health metrics to Digi Remote Manager. Added **Monitoring** > **Device Health** > **Only report changed values to Digi Remote Manager** option to control sending metrics to Digi Remote Manager on the basis of whether the values have changed since they were last reported. • Added **Monitoring** > **Device Health** > **Data point tuning** configuration options to fine tune what datapoints are uploaded as health metrics to Digi Remote Manager. • Added the ability to select Digi aView as the cloud service. ■ Added the ability to duplicate firmware to copy the active firmware to the secondary firmware partition. ■ Moved the **update firmware** CLI command to **system firmware update**. ■ Added new **Authoritative** option under TACACS+, RADIUS, and LDAP user authentication methods to prevent falling back to additional authentication methods. ■ Cisco Umbrella content filtering. ■ Added options under **System** > **Log** > **Server list** to allow users to specify the TCP/UDP protocol and port of the remote syslog server. ■ Enhanced SMS support: • Added **System** > **Scheduled tasks** > **Allow scheduled scripts** parameter to allow custom python scripts to handle sending/receiving SMS messages • Added the digidevice.sms python module for sending/receiving SMS messages in a custom python script. ■ MQTT client support via Paho Python module. ■ Added a random unprivileged port for performing ntp time syncs if standard port 123 fails. ■ Scripting enhancements:

Revision	Date	Description
		- Added a **Status** > **Scripts** page in the web UI and **show scripts** command to the Admin CLI to view custom scripts and applications configured in the device, along with their status. Added the **system scripts stop** command to the Admin CLI to stop a custom script or application. Additional updates: ■ Updated information about service and stand-alone modes. ■ Added information about updated AnywhereUSB Manager status information and AnywhereUSB Manager Hub connection messages. ■ Updated installer documentation.

Revision	Date	Description
G	December 2020	Release of Digi AnywhereUSB Plus firmware version 20.11: ■ Modem firmware update commands added to the Admin CLI. ■ Network bridging enhanced to use the MAC address of the first active device listed in **Network** > **Bridges** > ***Bridge name*** > **Devices** as the MAC address for the bridged interface. ■ IPsec enhancements: • Support for full IPsec IPv6 tunnels: IPv6-over-IPv4, or IPv4-over-IPv6 tunnels. • IPsec tunnels are now treated like network interfaces, which allows tunnels to be selected for routing and routing priority and access control lists. • IPSec tunnels now wait for Surelink tests, if configured, to pass prior to initiating outbound tunnels. ■ Policy-based routing enhancements: • Added a **DSCP** option to match the routing rule by the type of DSCP field in the packet. • Added a **Defaultroute** option for matching policy-based routes to the device's active default route. ■ Python pip support for installing external modules and libraries. ■ Link speed and duplex options added to Ethernet port configuration. ■ ssh command added to Admin CLI. ■ **Services** > **Ping responder** allows you to control the interfaces and firewall zones on which the DAL device will respond to ICMP requests. ■ Enhanced policy-based routing: • Added a **DSCP** option to match the routing rule by the type of DSCP field in the packet. • Added a **Defaultroute** option for matching policy-based routes to the device's active default route. ■ Added a link to User Guide under the User menu in the Web UI. Additional changes: ■ Updated latency graph information. ■ Added information for Always on top option.

Revision	Date	Description
H	February 2021	Updated the Get Started section. Updated the components list for all variants. Updated Console port section. Added information on how to define a static IP address. Added information about client ID length. Added Cannot uninstall the Manager from the Windows Apps screen.
J	March 2021	Release of Digi AnywhereUSB Plus firmware version 21.2: ■ Location services added, including: • The ability to define a static latitude and longitude as a location for the device. • Reporting location information as health metrics to Digi Remote Manager. • Geo-fencing: Allow you to define one or more circular or polygonal geo-fence areas and then perform a set of actions when the device enters or leaves that area. • Python support for location information through the digidevice.location python module. ■ Cellular modem carrier scanning and locking: • New modem scan CLI command for listing available carriers for the current modem and SIM. • Manual carrier selection option to allow you to lock the SIM to a specific carrier. ■ Enhanced serial support: • Certificate management control for TCP and autoconnect serial port setups. • Autoconnect. ■ Local REST API for automated configuration of the device. ■ Support for remote CLI commands through Digi Remote Manager. ■ Support for automatically checking for device and modem firmware updates. New features: ■ Right-click method to Add a Hub to the known Hub list. ■ Use all Hub addresses for connection to the Hub. ■ Cycle the power to a device connected to the Hub. ■ Create support log file. ■ Use the local REST API to configure the AnywhereUSB Plus device.

Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document "as is," without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Warranty

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

www.digi.com/howtobuy/terms

Customer support

Gather support information: Before contacting Digi technical support for help, gather the following information:

- ✓ Product name and model
- ✓ Product serial number (s)
- ✓ Firmware version
- ✓ Operating system/browser (if applicable)
- ✓ Logs (from time of reported issue)
- ✓ Trace (if possible)
- ✓ Description of issue
- ✓ Steps to reproduce

Contact Digi technical support: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Feedback

To provide feedback on this document, email your comments to

techcomm@digi.com

Include the document title and part number (AnywhereUSB® Plus User Guide, 90002383 A) in the subject line of your email.

Contents

AnywhereUSB® Plus User Guide

Get started

OPTIONAL: Use the CORE module to connect to the cellular network (AnywhereUSB 8 and 24 port devices ONLY)

Create groups and assign to client IDs

Connect to a group or USB device in the AnywhereUSB Manager

Manage the Hubs using the AnywhereUSB Manager

Configuration and management

Configure the AnywhereUSB in the web user interface

Interfaces

Console port

Services

User authentication

Firewall

System administration

Monitoring

Central management

Diagnostics

File system

Routing

Virtual Private Networks (VPN)

Command line interface

Configure the AnywhereUSB Manager from the command line

Security

Troubleshooting

Hardware

Regulatory and safety information

AnywhereUSB® Plus User Guide

AnywhereUSB® Plus is a Remote USB 3.1 Hub that implements USB over IP® technology over Gigabit Ethernet networks. The Hub enables communication with USB-enabled devices from virtualized systems and from remote host computers. You can securely deploy AnywhereUSB® Plus Remote USB 3.1 Hubs in non-secure environments, making it ideal for point-of-sale, kiosks, surveillance, industrial automation, or any mission-critical enterprise application. This Gigabit Ethernet-attached solution provides 2, 8, or 24 USB 3.1 ports to connect a wide range of peripheral devices such as USB license dongles, scanners, printers, cameras, storage media, or other USB devices. The 8- and 24-port models provide support for 10 Gigabit Ethernet and include SFP+ interfaces.

User roles

The different user roles that work with the AnywhereUSB Plus Hub are described in the table below.

Role	Description
Windows Administrator	The Windows administrators have the Windows permissions to install the **AnywhereUSB Manager** software on the computer. The Administrator can start, stop, and configure the **AnywhereUSB Manager** if it is run as a service.
Hub administrator	The Hub administrators have access to the Hub password. This enables the administrators to access and perform all activities to configure and maintain the Hub using the web UI and the Hub CLI commands.
User	A user can log into their computer and access the **AnywhereUSB Manager** that has been installed on the computer by the Windows administrator and is not running as a service. Within the **AnywhereUSB Manager**, the user can connect to the groups on the remote Hubs to which they have been given access by a Hub administrator. A user cannot access the web UI or use the Hub or AnywhereUSB Manager CLI commands.

Terminology

Role	Description
Computer	The physical or virtual equipment (such as a PC, laptop, or virtual machine), which is used to remotely access the AnywhereUSB Plus Hub.
Client ID	The client ID is a unique identifier assigned to a user account the first time a user logs in to a computer and opens the **AnywhereUSB Manager**. During this process, the **AnywhereUSB Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your user account with the Hub. For more information, see Client ID.
Group	A group is a set of USB ports on an AnywhereUSB Plus Hub with exclusive access to a single user account. Each USB port can be assigned to only one group by the Hub administrator. When you log into the computer and connect to a Hub, you are allowed to connect to any groups assigned to your client ID. See Create groups and assign to client IDs for more information.

Supported OS

Microsoft Operating systems supported:

- Microsoft Windows 7
- Microsoft Windows 8.1
- Microsoft Windows 10
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

Get started

Get started with your AnywhereUSB

This section explains what comes with each AnywhereUSB model, how to install the necessary software, and how to connect the hardware. After you have verified the AnywhereUSB Hub components, the software installation, hardware connection, and initial connection process must be done individually for each computer.

Initial connection: Administrators only

Note The steps in this process must be done by an administrator. Once the setup is complete, any user can connect to a group, as described below in Create and configure groups.

Step 1: Verify product components

Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone

Step 3: Install the AnywhereUSB Manager

Step 4: Connect the power supply

Step 5: Connect to the device using an Ethernet LAN connection

Step 6: Verify initial connection

Step 7: Change the admin password on the Hub

⚠ **WARNING!** If you do not replace the default password on the Hub for the default admin user, you are able to stage configuration changes, but **you will not be able to save the configuration changes**.

Step 8: Update the firmware on the AnywhereUSB

Next steps after initial connection

Step 9: Create and connect to groups

Step 10: Configure the Hub

Step 1: Verify product components

All AnywhereUSB models include the AnywhereUSB device in the box. Additional equipment may be required or may be optional.

- AnywhereUSB 2 Plus components
- AnywhereUSB 8 Plus components
- AnywhereUSB 24 Plus components

NEXT STEP: If you are performing the initial device set-up, proceed to the next step after verifying the components: Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone.

AnywhereUSB 2 Plus components

Verify that you have the following included and required additional equipment.

Included equipment

Equipment	Description
AnywhereUSB 2-port device	For details, see AnywhereUSB 2 Plus: Front panel.
Loose label sticker	A loose label sticker that includes the unique device password is included in the box. This default password will be needed if the device is factory reset and you want to access the web UI on the device. Retain this label sticker with your hardware records. See QR code definition for information about the information contained in the QR code.

Required additional equipment

Equipment	Description
Ethernet cable	STP Cat 7 Ethernet cable. See Step 5: Connect to the device using an Ethernet LAN connection.
Power supply kit	Recommended item: 1.8 amps per port. Digi PN 76000965. See Step 4: Connect the power supply.
Alternate power supply kits	These may be used instead of the recommended power supply kit if USB port charging is not required: ■ AC Power Supply: US plug to 5 VDC. 2.5 mm locking barrel plug (3 A max). Digi PN 76000934. ■ AC Power Supply: EU plug to 5 VDC. 2.5 mm locking barrel plug (3 A max). Digi PN 76000935. ■ AC Power Supply: Standard Temperature, Universal plugs (US, EU, UK, AU) to 5 VDC. 2.5 mm locking barrel plug (3 A max). Digi PN 76002021.

Optional additional equipment

DIN rail mounting kit	Digi PN 7000682. See Attach a DIN rail clip (AnywhereUSB Plus 2-port ONLY). **Note** Some kits may not have the required screws included. If this occurs, you will need to separately purchase two screws of the following type: 4-40 x .250 Flat head, Phillips head, zinc-plated screws.

AnywhereUSB 8 Plus components

Verify that you have the following included and required additional equipment. A list of optional equipment is also included below.

Included equipment

Equipment	Description
AnywhereUSB 8-port device	For information about the hardware, see: ■ AnywhereUSB 8 Plus: Front panel ■ AnywhereUSB 8 Plus: Back panel
Power supply	Connect the power supply to the Hub and tighten the screws to secure. See Step 4: Connect the power supply.
Rack mounting brackets and screws	Use these items to mount the device onto a server rack.
Loose label sticker	A loose label sticker that includes the unique device password is included in the box. This default password will be needed if the device is factory reset and you want to access the web UI on the device. Retain this label sticker with your hardware records. See QR code definition for information about the information contained in the QR code.

Required additional equipment

Equipment	Description
Ethernet cable	STP Cat 7 Ethernet cable See Step 5: Connect to the device using an Ethernet LAN connection.
Power cord	IEC power cord See Step 4: Connect the power supply.

Optional additional equipment

Equipment	Description
SFP+ module	Connect an SFP transceiver module for fiber connection, such as Finisar Network FTLX8574D3BCL SFP+.
Console cable	RS232 DB9 Console cable Use the console cable to establish a serial connection from the serial port on your device to your local laptop or PC. See Console port.

Optional additional equipment for connecting to a cellular network

This equipment is required only if you want to connect to a cellular network. See OPTIONAL: Use the CORE module to connect to the cellular network (AnywhereUSB 8 and 24 port devices ONLY).

Equipment	Description
Digi CORE® module	
SIM card	An activated SIM card provided by your cellular network operator. You can insert up to two SIM cards in the CORE module. The CORE module supports the standard mini-SIM cards (2FF).
Antennas (2)	

AnywhereUSB 24 Plus components

Verify that you have the following included and required additional equipment. A list of optional equipment is also included below.

Note The power supply for the AnywhereUSB 24 Plus is built into the device.

Included equipment

Equipment	Description
AnywhereUSB 24-port device	For more information, see: ■ AnywhereUSB 24 Plus: Front Panel ■ AnywhereUSB 24 Plus: Back panel
Loose label sticker	A loose label sticker that includes the unique device password is included in the box. This default password will be needed if the device is factory reset and you want to access the web UI on the device. Retain this label sticker with your hardware records. See QR code definition for information about the information contained in the QR code.

Required additional equipment

Equipment	Description
Ethernet cable	STP Cat 7 Ethernet cable See Step 5: Connect to the device using an Ethernet LAN connection.
Power cord	IEC power cord See Step 4: Connect the power supply.

Optional additional equipment

Equipment	Description
Additional Ethernet cable	STP Cat 7 Ethernet cable

Equipment	Description
Additional power cord	IEC power cord See Step 4: Connect the power supply. **Note** Digi recommends that you purchase an additional power cord for the following reasons: **More power is needed if you use all 24 ports. **If you do not use all 24 ports, two power cords maintain redundancy if one power supply fails. Digi also recommends plugging each power cord into separate main power circuits. **Helps maximize heat dissipation.
SFP+ modules	Connect an SFP transceiver module for fiber connection, such as Finisar Network FTLX8574D3BCL SFP+.
Console cable	RS232 DB9 Console cable Use the console cable to establish a serial connection from the serial port on your device to your local laptop or PC. See Console port.

Optional additional equipment for connecting to a cellular network

This equipment is required only if you want to connect to a cellular network. See OPTIONAL: Use the CORE module to connect to the cellular network (AnywhereUSB 8 and 24 port devices ONLY).

Equipment	Description
Digi CORE® module	
SIM card	An activated SIM card provided by your cellular network operator. You can insert up to two SIM cards in the CORE module. The CORE module supports the standard mini-SIM cards (2FF).
Antennas (2)	

Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone

You can choose to install the **AnywhereUSB Manager** in service or stand-alone mode. Each mode offers different features and may interact differently with the **Manager**.

Note The **AnywhereUSB Manager** shows information that pertains to the installed mode. Most importantly, if you install in service mode, "AnywhereUSB SERVICE MODE" displays as the **Manager** title. See AnywhereUSB Manager Status pane for detailed information.

The table below compares the features in each mode. Refer to the table to help you determine which mode is best for your organization.

Feature	Service mode	Stand-alone mode
Run and configure the AnywhereUSB Manager	Only an Administrator can run the **AnywhereUSB Manager** to configure the service.	Any user (an Administrator or a non-Administrator) can run and configure the **AnywhereUSB Manager**.
USB device availability	The devices in the groups connected to the computer are always available to the computer. The service automatically runs in the background. Note To ensure that all USB devices are connected to your computer at boot time, you must select Enable Auto Connect for each group assigned to the client ID for the computer.	Devices in connected groups are only available when the **Manager** is running.
Which users can see devices connected to the computer	All users can see all the devices in the groups that are connected to the computer.	All users can see all the devices in the groups that are connected to the computer. Note The devices that can be seen are changeable, depending on which users are logged into the computer.

Mode interactions with AnywhereUSB features

The sections below explain how each mode interacts with the **AnywhereUSB Manager** features.

Service

- To ensure that all USB devices are connected to your computer at boot time, you must select Enable Auto Connect for each group assigned to the client ID for the computer. The USB devices in the groups connected to the computer are available to the users.
- Multiple users can log on with their Windows user account and use the devices connected by the service to the computer at the same time.
- If you are not an Administrator, you are able to see and use the devices that are in the in the connected groups, but you cannot choose to connect to an additional Hub or device.
- Groups and devices remain connected when users log in or out.

Stand-alone

- If you install the **AnywhereUSB Manager** as a stand-alone, Digi recommends that you select the **Run AnywhereUSB Manager at Startup** option during the installation process to automatically launch the **Manager** each time you log in to your Windows user account.
- When the user logs in and starts the **AnywhereUSB Manager**, the **Manager** automatically connects to groups that have Enable Auto Connect enabled. The USB devices in those groups are connected to the machine.
- Groups and devices are connected when the **Manager** starts running if auto connect is enabled for the group. If auto connect is not enabled for the group, you can manually connect to a group. Groups and devices are disconnected when the **Manager** stops running, which typically occurs when the user running the **Manager** logs off the computer.

Warnings

- Only an Administrator has the rights to install the **AnywhereUSB Manager**.

  If you log onto the computer as a non-Administrative user and attempt to install the **AnywhereUSB Manager**, you will be prompted during the installation process for an Administrator user name and password. If you do not provide Administrator credentials, you will not be able to complete the installation process.
- In stand-alone mode, only one user can open the **AnywhereUSB Manager** at a time. The **Manager** cannot be opened simultaneously by multiple users. In addition, a single user cannot run multiple instances of the **Manager**.
- In stand-alone mode, each user must have a different client ID, which results in an individual **Manager** configuration. Digi does not support sharing a client ID between two different Windows users or computers.
- Digi recommends that you do NOT install the **AnywhereUSB Manager** as a stand-alone, re-install it, and then choose to run the **Manager** as a service. If this does occur, be aware that the stand-alone and the service will have separate configurations. The **Manager** or service will only use the stand-alone or service configuration, respectively.
- If you install the **Manager** as a service and then stop the service, the **AnywhereUSB Manager** will choose not to run.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 3: Install the AnywhereUSB Manager.

Step 3: Install the AnywhereUSB Manager

The **Anywhere USB Manager** software must be downloaded from the Digi support site and installed on your computer. After the manager software installs, the **AnywhereUSB Manager** launches. The **AnywhereUSB Manager** automatically discovers AnywhereUSB Hubs on the local subnet.

CAUTION! Only a Windows Administrator can perform the software install. If you are logged in as a non-Windows Administrator user and you attempt to install the software, you will be required to enter Windows Administrator login credentials to be able to complete the installation process.

Prerequisites

Before you begin, you should decide whether you want to run the **AnywhereUSB Manager** as a stand-alone or as a service. For detailed information, see Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone.

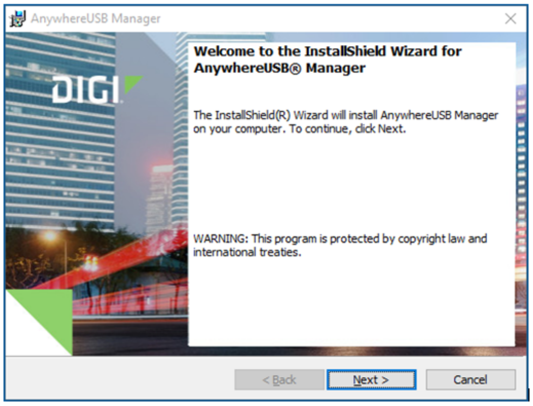
1. Download the AnywhereUSB Manager installer from the AnywhereUSB support page.

   Note This link takes you to the AnywhereUSB 2 Plus drivers page, but the driver options are the same for all AnywhereUSB Plus models.

2. Select and download the appropriate software for your operating system.

   Note You can save the downloaded software to your computer before you start the install process. This is useful if you decide to uninstall the **AnywhereUSB Manager** in the future.

3. Right-click on the downloaded software and select the **Run as Administrator** menu option.
4. Enter your Administrator login credentials. The **AnywhereUSB Manager** installation wizard launches.



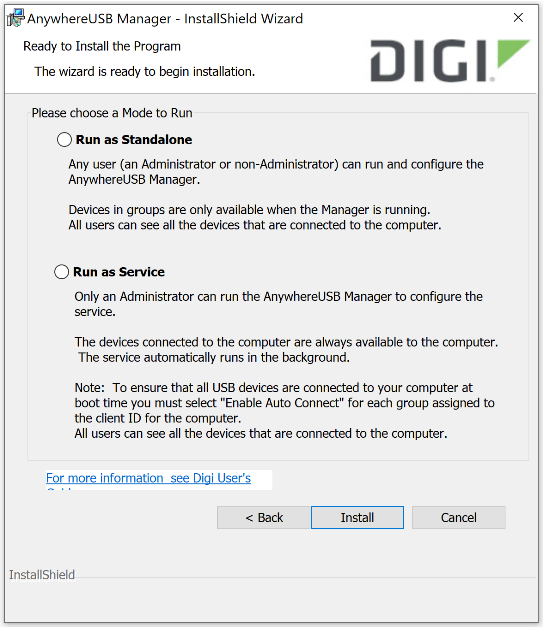
Welcome to the InstallShield Wizard for AnywhereUSB® Manager

The InstallShield(R) Wizard will install AnywhereUSB Manager on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

7. Click **Next**. The **Ready to Install** screen appears.
8. You must specify which mode you want to install: Standalone or Service. For detailed information about each mode, refer to Step 2: Determine how to run AnywhereUSB Manager:

Service or stand-alone.



AnywhereUSB Manager - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

Please choose a Mode to Run

○ **Run as Standalone**

Any user (an Administrator or non-Administrator) can run and configure the AnywhereUSB Manager.

Devices in groups are only available when the Manager is running.
All users can see all the devices that are connected to the computer.

○ **Run as Service**

Only an Administrator can run the AnywhereUSB Manager to configure the service.

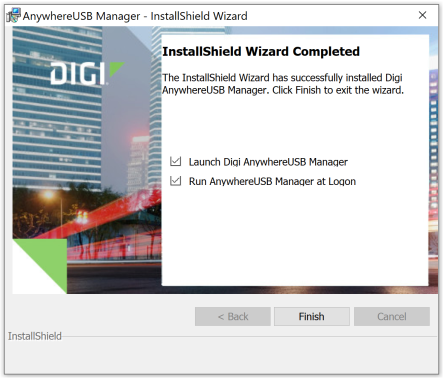
The devices connected to the computer are always available to the computer.
The service automatically runs in the background.

Note: To ensure that all USB devices are connected to your computer at boot time you must select "Enable Auto Connect" for each group assigned to the client ID for the computer.
All users can see all the devices that are connected to the computer.

For more information see Digi User's

9. Click **Install**. A status bar shows the progress of the installation process. When complete, the **Completed** screen appears.
10. The options in the **Completed** screen are selected by default. Click the checkbox if you do not want to use the feature.
    - **Launch AnywhereUSB Manager**: Launches the **AnywhereUSB Manager** when the installation completes.
    - **Run AnywhereUSB Manager at Logon**: Automatically launch **AnywhereUSB Manager** each time you log in to your Windows user account. Digi recommends that you do not de-select this option.

Note If you have installed the **Manager** as a service, this option applies only to the current admin user. Each time this admin user logs in, the **Manager** launches so the user can administer the service. If a non-admin user logs in, the service is available, but the **AnywhereUSB Manager** does not display.



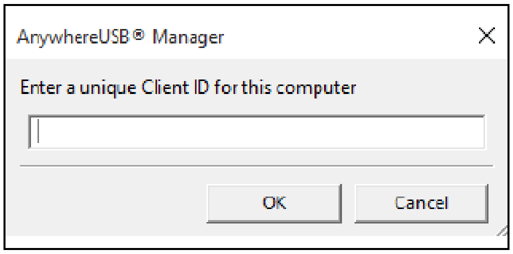
AnywhereUSB Manager - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed Digi AnywhereUSB Manager. Click Finish to exit the wizard.

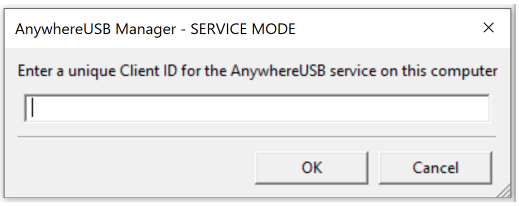
☑ Launch Digi AnywhereUSB Manager
☑ Run AnywhereUSB Manager at Logon

11. Click **Finish**. The client ID confirmation dialog appears.
    - **Stand-alone**: If you installed the **Manager** in stand-alone mode, the client ID confirmation dialog looks like this:


AnywhereUSB® Manager

Enter a unique Client ID for this computer

    - **Service**: If you installed the **Manager** in service mode, the client ID confirmation dialog looks like this:


AnywhereUSB Manager - SERVICE MODE

Enter a unique Client ID for the AnywhereUSB service on this computer

12. Enter a unique client ID. This client ID is associated with the login credentials for the user currently logged on to the computer. See Client ID for more information about how the client ID is used by your computer and the Hub to create a connection.
13. Click **OK**. If the **Launch AnywhereUSB Manager** was selected, the **AnywhereUSB Manager** launches.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 4: Connect the power supply.

Step 4: Connect the power supply

This section explains how to power the Hub.

Before you begin, verify that you have your AnywhereUSB Hub and the required additional equipment. See Step 1: Verify product components.

1. Connect the appropriate power supply for your model to the device.
    - **AnywhereUSB 2 Plus Hub**: Connect the power supply kit to the Hub, and use the twist-lock feature to secure the power supply to the Hub.
    - **AnywhereUSB 8 Plus Hub**: Connect the power supply to the Hub and tighten the screws to secure. Connect the power cord to the power supply.
    - **AnywhereUSB 24 Plus Hub**: Connect both IEC power cords into the Hub. Note that the power supply is built into the Hub.

    Note Digi recommends that you purchase an additional power supply for the following reasons:
    **More power is needed if you use all 24 ports.

**Two power cords maintain redundancy if one power supply fails. Digi also recommends plugging each power cord into separate main power circuits.
**Helps maximize heat dissipation.

2. Plug the power cord into an outlet.

   Note For an AnywhereUSB 24 Plus Hub, plug both power cords into an outlet, if you are using two power cords. Digi recommends plugging each power cord into separate main power circuits.

3. Verify that the blue power LED is illuminated.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 5: Connect to the device using an Ethernet LAN connection.

Step 5: Connect to the device using an Ethernet LAN connection

Connect an Ethernet cable to your PC and Hub to create an Ethernet LAN network. This enables you to access the Hub's web UI and configure the Hub.

⚠ **WARNING!** Digi recommends that you use a private network to connect the computer to the Hub. This ensures that only clients IDs with known user credentials can connect to the Hub. The first time that a client ID on a computer connects to the Hub, the unique credentials for this known user are stored in your Hub. If do not use a private network, an unknown computer with the same client ID may happen to connect to the Hub before the known computer connects. In this case, the known computer will not be able to connect and authenticate.

1. Connect one end of a Shielded CAT 7 (STP) Ethernet cable to the ETH port on the Hub.
2. Connect the other end of the Ethernet cable to your computer.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 6: Verify initial connection.

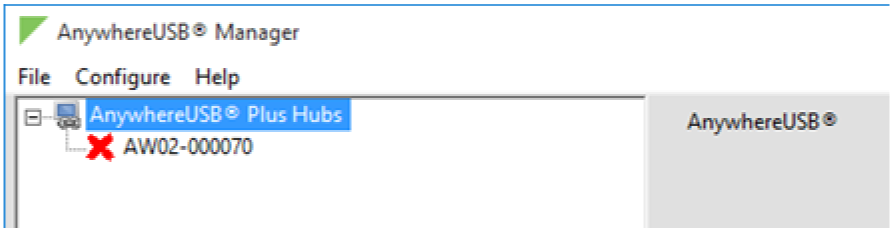
Step 6: Verify initial connection

After the hardware has been connected and powered on, and you have installed the **AnywhereUSB Manager**, verify that the Hub connection is working as expected.

Note You will need a USB flash drive to follow the verification process below.

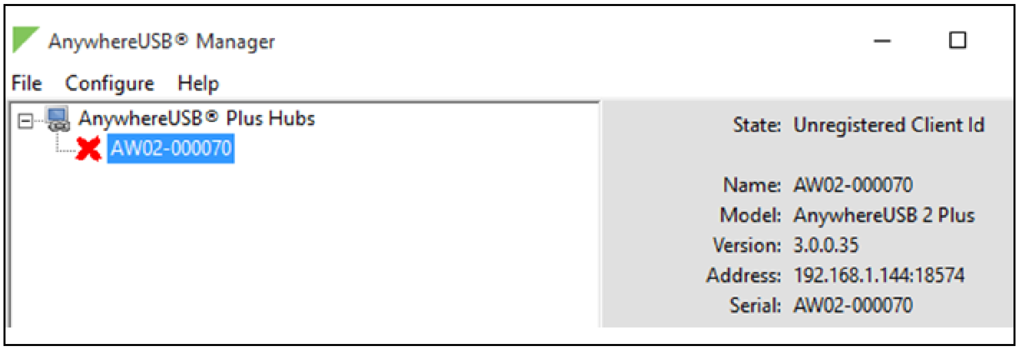
1. Verify that your Hub powered on. The power LED is solid blue.
2. Plug your USB flash drive into port 1 on the Hub.
3. Verify that the USB port 1 LED is solid yellow, green, or blue, depending on whether the USB flash drive is 1.1, 2.0, or 3.1.

4. If not already open, launch the AnywhereUSB Manager.
5. Expand **AnywhereUSB Hubs** to display a list of AnywhereUSB Hubs.



6. Verify that the serial number of the Hub is in the list. You can find the serial number on the Hub's label.
7. You will notice that the Manager is showing the Hub in an error state, with a red X appearing next to the Hub name. Click on the Hub to update information in the **Hub Status** pane. The Hub **State** appears as "Unregistered Client ID."

   This is a security feature. The Hub administrator needs to allow each new client ID by adding the client ID to the client list.



State: Unregistered Client Id

Name: AW02-000070
Model: AnywhereUSB 2 Plus
Version: 3.0.0.35
Address: 192.168.1.144:18574
Serial: AW02-000070

8. Before you can register the client ID with the Hub, you must add the client ID to the Hub from the web UI.
   a. Right-click on the Hub and select **Open Web UI**.
   b. A login dialog displays. Enter the following:
      - **User name**: admin
      - **Password**: Located on the label on the bottom of the Hub. Note that the password is case-sensitive and must be typed in exactly as it appears on the label.

   Note The first time you launch the web UI, a warning dialog may appear if your internet connection is not private. In this situation, continue to access the device. The log in dialog appears.

   c. Click **Login**. The web UI appears.
   d. You are required to change the password the first time you log in. See Change the default password for the admin user.
   e. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
   f. Expand the **Client Settings** section.

g. Click **Add Client**. A new row labeled "New Client" is added to the client list and the **Settings for Client** section is populated for the new client.

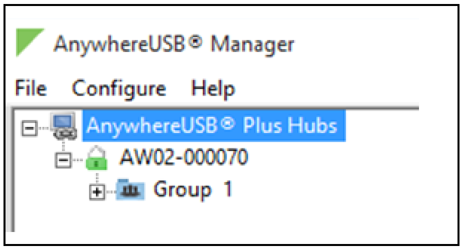
h. In the **Client ID** field, enter the client ID you assigned to your user login credentials.

i. In the **Description** field, enter a descriptive name for the computer. This step is optional.

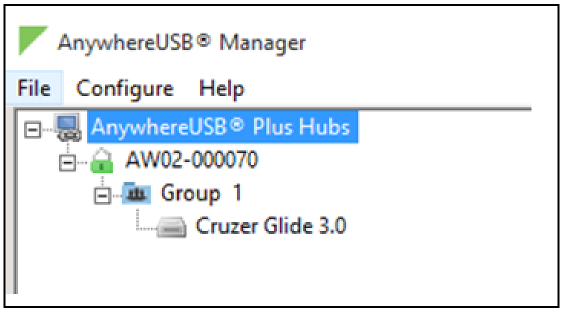
j. Click the check box next to group **1**.

k. Click **Apply** to save the Hub settings.

9. Open the **AnywhereUSB Manager**. The **Manager** connects to the Hub.
10. Expand the Hub to display the groups.



11. Expand Group 1 to display the USB flash drive connected to Group 1.



12. Right-click on Group 1 and select **Connect to Group**. The USB flash drive is available in Windows.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 7: Change the admin password on the Hub.

Step 7: Change the admin password on the Hub

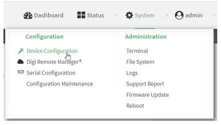
After initial installation, you must change the admin password on the Hub for the default **admin** user. This ensures that when you log into the WebUI or the command line you are able to save any configuration changes you have made.

WARNING! If you do not replace the default password on the Hub for the default admin user, you are able to stage configuration changes, but **you will not be able to save the configuration changes**.

The unique, factory-assigned password for the default **admin** user account is printed on the bottom label of the device and on the loose label included in the package. When you first log into the WebUI, you will be required to change the password for the **admin** user prior to being able to save any configuration changes.

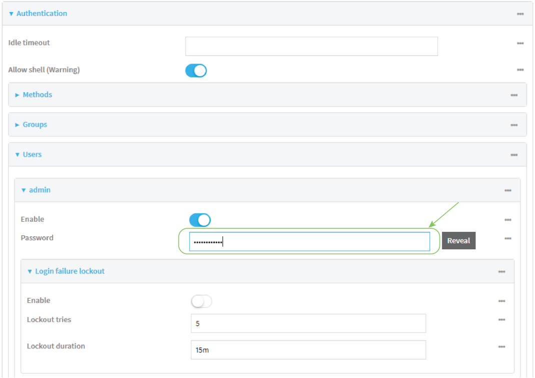
If you erase the device configuration or reset the device to factory defaults, the password for the **admin** user will revert to the original, factory-assigned default password.

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

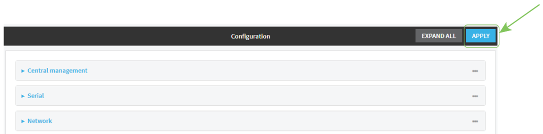
3. For **Password**, enter the new password. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.



Lockout tries: 5

Lockout duration: 15m

4. Click **Apply** to save the configuration and apply the change.



For more detailed information about this process, see Change the default password for the admin user.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step: Step 8: Update the firmware on the AnywhereUSB.

Step 8: Update the firmware on the AnywhereUSB

You should update the firmware on the device to ensure that you have the latest features.

1. Get the latest version of the firmware.
   a. Navigate to the **Firmware Updates** section of the AnywhereUSB support page for your variant:
      - AnywhereUSB 2 Plus
      - AnywhereUSB 8 Plus
      - AnywhereUSB 24 Plus

   b. Download the firmware onto your computer, and make note of the location.
2. Log into the AnywhereUSB Web UI as a user with Admin access.
3. On the main menu, click **System**. Under **Administration**, click **Firmware Update**.
4. Click **Upload file**.
5. Click **Choose File**. A dialog appears.
   a. Navigate to the location to which you downloaded the firmware file.
   b. Select the file.
   c. Click **Open**.
6. Click **Update Firmware**.

For more detailed information about this process, see Update system firmware.

NEXT STEP: If you are performing the initial device set-up, you have now completed all of the required steps. You can return to Get started for information about steps after the initial connection: Step 9: Create and connect to groups and Step 10: Configure the Hub.

Step 9: Create and connect to groups

After you have completed your initial connection, you can create groups and assign ports to each group. Once this step is complete, you can specify the groups that a client ID is allowed to access.

- Create groups and assign to client IDs
- Connect to groups.

  You can open the **AnywhereUSB Manager** and connect to the Hubs and groups to which access is allowed. The type of user (Administrator or non-Administrator) that can open the **Manager** depends on the installation method you selected.

  - **Stand-alone**: Any user (an Administrator or a non-Administrator) can run the **AnywhereUSB Manager**.
  - **Service**: Only an Administrator can run the **AnywhereUSB Manager**.

NEXT STEP: If you are performing the initial device set-up, proceed to the next step after initial connection: Step 10: Configure the Hub.

Step 10: Configure the Hub

The Hub administrator can use the web UI to configure networks parameters, services, and other Hub features. You can update the firmware, back up the configuration, view system information and logs, and reboot the Hub. To get started, see Configure the AnywhereUSB in the web user interface.

NEXT STEP: If you are performing the initial device set-up, you have now completed all of the steps. You can return to Get started.

OPTIONAL: Use the CORE module to connect to the cellular network (AnywhereUSB 8 and 24 port devices ONLY)

This section explains how to connect the CORE module and cellular antennas to the AnywhereUSB hardware. You can then connect to a cellular network to connect to a support management tool, such as Digi Remote Manager.

You must have purchased a CORE module to be able to connect to the cellular network.

Note This section applies to AnywhereUSB 8 and 24 port devices ONLY. You cannot configure a cellular network connection for an AnywhereUSB 2 device.

Prerequisites

- Activated SIM card(s) from your cellular network provider. Up to two SIM cards can be inserted into the CORE module.
- CORE module. This may be included with your device. If it is not, you must purchase one separately.

Connect the hardware and connect to the cellular network

1. Insert your activated SIM card (or cards) into the CORE module. The notched end of SIM card should be inserted first, with the gold metal contacts facing down. You will hear a click once the SIM is completely inserted.

   Note If one SIM card is being used, insert the SIM card into the SIM 1 slot.

2. Insert the CORE module into the device.
   a. Orient the device so the rear of the device is facing you.
   b. Remove the CORE module slot cover from the back of the device.
   c. Insert the CORE module into the slot. Make sure the pin holes on the back of the module match the location of the pins in the slot.
   d. Push the module into the slot.
   e. Push the white handle in until you hear it click.
   f. Optionally, you can screw one of the CORE module cover screws into the center of the handle.

   g. Place the CORE module slot cover over the CORE module. Make sure that the antenna labels are oriented correctly.
   h. Use the two screws to attach the CORE module slot cover to the device.

3. Attach both of the antennas included with the CORE module equipment. While gripping the metal connector section with your thumb and forefinger, tighten until secure. Do not tighten the antenna by holding any part of the plastic antenna housing.

   Note Attaching both antennas ensures maximum performance. If a single antenna solution is required, it must be attached to the antenna port labeled MAIN.

4. Connect the appropriate power supply for your model to the device.
   - **AnywhereUSB 8 Plus Hub**: Connect the power supply to the Hub and tighten the screws to secure.
   - **AnywhereUSB 24 Plus Hub**: Connect both IEC 60320 power supplies into the Hub.

     Note Digi recommends that you purchase an additional power supply for the following reasons:
     **More power is needed if you use all 24 ports.
     **Two power cords maintain redundancy if one power supply fails. Digi also recommends plugging each power cord into separate main power circuits.
     **Helps maximize heat dissipation.

5. Plug the power supply to an outlet.

   Note For an **AnywhereUSB 24 Plus Hub**, plug both power supplies into an outlet, if you are using both power supplies. Digi recommends plugging each power cord into separate main power circuits.

Create groups and assign to client IDs

For each Hub, the Hub administrator can assign a number of USB ports to a group. The Hub administrator can also assign groups to client IDs.

When the client ID connects to a Hub, the computer is allowed to access the ports in the groups assigned to the client ID. The same groups can be assigned to more than one client ID on a Hub. Note that connecting to a group is exclusive and only one client can connect to a group at a time.

Groups are created and assigned to client IDs in the **AnywhereUSB** page in the web UI.

1. Create groups and assign ports to the group.
2. Assign a group to a client ID.

Create groups and assign ports to the group

In the **AnywhereUSB** page in the web UI, you can assign a name to each group, and specify the USB ports in each group. Each port can only be assigned to one group. Any unassigned ports are automatically included in the **Unassigned** row that displays beneath the list of groups.

- If a group has ports assigned to it, the group will display in the **AnywhereUSB Manager**, even if a USB device is not connected to a port.
- If you don't want a group with all unused USB ports to appear in the the **AnywhereUSB Manager**, you can reassign the unused ports in a group to a different group. See Hide a group in the AnywhereUSB Manager.

To create a group and assign USB ports to the group:

1. Open the web UI for your selected Hub.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Expand the **Group Settings** section.
4. In the **Group Description** field, enter a name for a group. This name displays in the **Group Name** field in the Group Status pane in the **Anywhere USB Manager**.
5. In the row for the group, select the ports for that group. Each port on a Hub can be assigned to only one group. Ports that are not assigned to a group can be put in the default **Unassigned** group.
6. Repeat the steps 4 and 5 for each group you want to create.
7. When done, click **Apply** to save the changes.

Assign a group to a client ID

You can assign the groups to a client ID. When the client ID connects to the Hub, the computer can access all of the ports in the specified groups.

Note Make sure that you have at least one client ID created for the Hub. You can manually add client IDs, if needed. See Add client IDs to the client list.

1. Open the web UI.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Expand the **Client Settings** section.
4. In the client list, select the client ID to which you want to assign groups. Information about the selected client ID displays in the **Settings for Client** section.
5. Click the check box next to a group to which the computer is allowed access. As you select groups, the selected group numbers appear in the **Group Access** field in the **Settings for Client** section.

   You can also manually enter group numbers in the **Group Access** field.
6. Click **Apply** to save the changes.

Connect to a group or USB device in the AnywhereUSB Manager

When you connect to a group, you are given exclusive access to all of the USB ports in the group to which you are allowed access. All other users are blocked from access to the ports in that group until you disconnect from the group. A user can connect to more than one group at a time.

When a USB device is plugged in to a port on a Hub, the device displays in the list of devices in the group. Note that a group may have ports that do not have a connected device. Only ports with a connected USB device display in the **AnywhereUSB Manager**.

Auto connect enabled for a group

If you have enabled auto connect for a group, you are automatically connected to those groups when:

- You log in to your computer and **AnywhereUSB Manager** opens automatically
- You manually open and log into **AnyhwereUSB Manager**.
- If the **Manager** is running as a service.

See Configure auto connect for more information.

Note When you open the **AnywhereUSB Manager**, the **Manager** attempts to connect to the groups to which you are allowed access. If someone else already owns the group, you will not be connected to that group.

Connect to a group or a USB device in the AnywhereUSB Manager

You can connect to all of the USB devices and ports in a group, or to one device in a group.

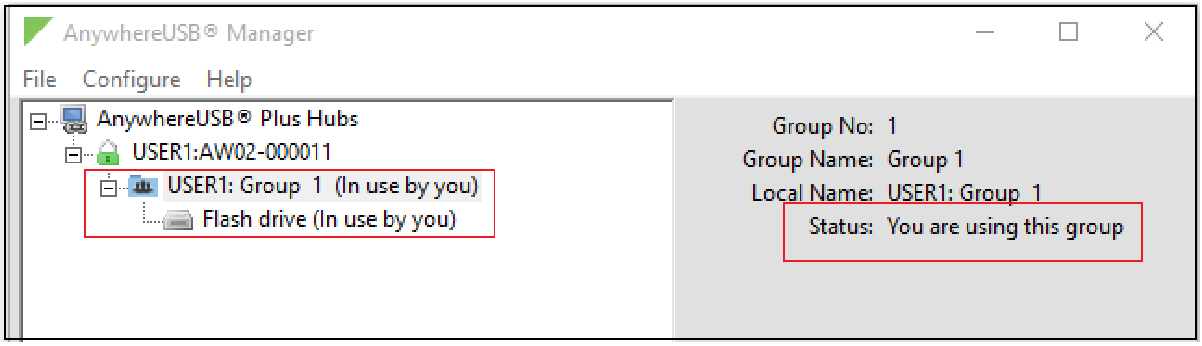
- **Connect to a group**: To connect to a group, right-click on the group name and click Connect to Group.
- **Connect to USB ports in a group**: You can connect to the USB ports in a group depending on whether you are allowed access to the port and if you are connected to the group:
    - If you are connected to the group, right-click on a USB device name and click Connect to Device. You are connected to that USB device and to all of the USB ports in the group.
    - If you are not connected to the group, right-click on the USB device name and click Connect to Group to connect to the group and the USB device.
    - If the group is owned by another user, you are not allowed to connect to the device.

Connect to a group

You can connect to a group so that you have access to the ports in the group. Once you have connected to a group, no one else can connect to that group. You cannot connect to a group that is already is use.

When you have connected to a group, a note appears next to the group name, next to the devices in the group, and in the Group Status pane to show that the device is being used by you.

1. Open the **Anywhere USB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Right-click on the group to which you want to connect.
5. Select **Connect to Group**. A note appears next to the group name, next to the devices in the group, and in the Group Status pane to show that the device is being used by you.



Group No: 1
Group Name: Group 1
Local Name: USER1: Group 1
Status: You are using this group

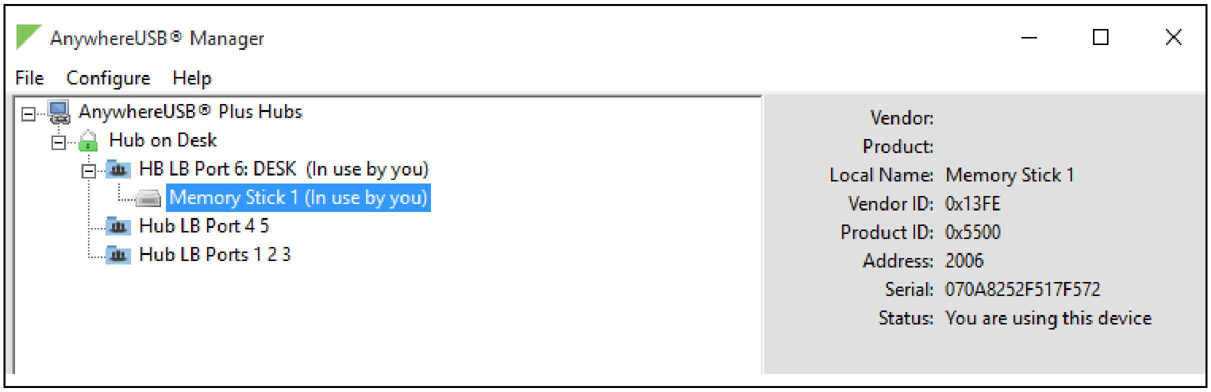
Connect to a USB device

You can connect to a USB device in a group to which you have access. You cannot connect to a device in a group that is already is use by another user.

When you have connected to a device, a note appears next to the device name and in the Device Status pane to show that the device is being used by you.

1. Open the **Anywhere USB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Expand a group to display the USB devices in the group.
5. Right-click on the device to which you want to connect.
6. The connect menu option available depends on whether you are already connected to the group.
   - **Connected to the group**: Right-click on the USB device name and click **Connect to Device** to connect to the USB device.
   - **Not connected to the group**: Right-click on the USB device name and click **Connect to Group** to connect to the group and the USB device.

   A note appears next to the device name and in the Device Status pane to show that the device is being used by you.



Vendor:
Product:
Local Name: Memory Stick 1
Vendor ID: 0x13FE
Product ID: 0x5500
Address: 2006
Serial: 070A8252F517F572
Status: You are using this device

Manage the Hubs using the AnywhereUSB Manager

You can use the **AnywhereUSB Manager** to view the AnywhereUSB Plus Hubs that are allowed to connect to your computer. You can also connect to groups of USB ports on the Hubs.

By default, the **AnywhereUSB Manager** is configured to automatically discover Hubs that are connected to the same network as your computer. You can allow a connection to additional Hubs that are not on the same network.

Note Before you begin, make sure you have installed the **AnywhereUSB Manager**.

Note The **AnywhereUSB Manager** supports the AnywhereUSB Plus family of products: AnywhereUSB 2 Plus, AnywhereUSB 8 Plus, AnywhereUSB 24 Plus. The earlier AnywhereUSB products (AnywhereUSB 2, AnywhereUSB 5, and AnywhereUSB 14) use a different driver package. For more information, please refer to the AnywhereUSB product page.

Launch the AnywhereUSB Manager

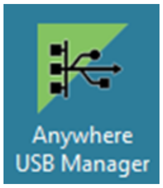
You can search for and launch the **Anywhere USB Manager** using the Windows application search feature or from the Digi International **Start** menu.

Note If the **Anywhere USB Manager** was configured during the installation process to automatically launch when you logged in, you do not need to do this step.

Installed in standalone mode

To manually start the **Anywhere USB Manager**:

1. Log in to your computer.
2. Double-click the **Anywhere USB Manager** shortcut on your desktop.



Rename AnywhereUSB Hubs, groups, and USB devices

Each AnywhereUSB Hub and group has a default name that displays in the **AnywhereUSB Manager**. You can also assign a local name to each Hub, group, or USB device that displays in the **AnywhereUSB Manager**, which can help you to uniquely identify your local Hubs, groups, and USB devices.

The local name is local to the computer on which the **AnywhereUSB Manager** is running. No other user can see the local name.

- Assign a local name to a Hub
- Assign a local name to a group
- Assign a local name to a USB device

Assign a local name to a Hub

You can give a AnywhereUSB Hub a local name. The name displays in the Hub Status pane in the **AnywhereUSB Manager** and also in the tree view. The local name is local to the computer on which the **AnywhereUSB Manager** is running.

Note The Hub local name is different from the default Hub name. For detailed information about the default name, see Rename a Hub and the groups in a Hub.

1. Open the **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.

3. Right-click on the Hub that you want to give a local name.
4. Select the **Assign Local Name** menu option. A dialog appears.
5. In the field, enter a local name for the Hub.
6. Click **OK**.

Assign a local name to a group

You can give a group a descriptive local name. The local name displays in the Group Status pane in the **AnywhereUSB Manager** and also in the tree view. The local name can be seen only on the computer on which the **AnywhereUSB Manager** is running.

Note The group local name is different from the default group name. For detailed information about the default name, see Rename a Hub and the groups in a Hub.

1. Open the **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand the Hub that has the group you want to give a local name.
4. Right-click on the group that you want to rename.
5. Select the **Assign Local Name** menu option. A dialog appears.
6. Enter a local name for the group.
7. Click **OK**.

Assign a local name to a USB device

You can assign a local name to a USB device that displays in the in the Device Status pane and also in the tree view. The local name is local to the computer on which the **AnywhereUSB Manager** is running.

1. Open the **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand the Hub that has the group to which the to USB device is attached.
4. Expand the appropriate group to display the USB devices in the group.
5. Right-click on the USB device that you want to give a local name.
6. Select the **Assign Local Name** menu option. A dialog appears.
7. In the field, enter a local name for the USB device.
8. Click **OK**.

Disconnect from a group or USB device

You can disconnect from any group or USB device in the group to which you no longer need access.

- Disconnect from a group
- Disconnect from a USB device

Disconnect from a group

You can disconnect from a group that has ports you no longer need access to. You are disconnected from all USB devices and ports in that group. Any other user can then connect to that group.

Note If you have auto connect enabled for the group, you are not allowed to disconnect from the group. You have to first disable auto connect, and then disconnect from the group. The next time you log in to your computer, you will not be automatically connected to this group.

1. Open **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Right-click on the AnywhereUSB group from which you want to disconnect.
5. Select **Disconnect from Group**. A note appears in the Group Status pane to show that the group is not being used.

Disconnect from a USB device

You can disconnect from a USB device to which you no longer need access. You can disconnect from a USB device that is in a group to which you are connected. Other users cannot connect the USB device, since you still own the group that the USB device is in.

Note If you have auto connect enabled for the group, you can disconnect from a USB device in the group, but note that the device will be connected again the next time you log in to your account.

1. Open **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Expand a group to display the USB devices in the group.
5. Right-click on the USB device from which you want to disconnect.
6. Select **Disconnect from Device**. A note appears in the Device Status pane to show that the device is not being used.

Configure auto connect

You can enable the auto connect feature for a group (or multiple groups). This feature ensures that whenever you open the **AnywhereUSB Manager**, you are automatically connected to all of the groups to which you are allowed access that have auto connect enabled.

Ways that you can open the **AnywhereUSB Manager**:

- You log in to your computer and **AnywhereUSB Manager** opens automatically.
- You manually open and log into **AnywhereUSB Manager**.
- If the **Manager** is running as a service.

Note When you open the **AnywhereUSB Manager**, the **Manager** attempts to connect to the groups to which you are allowed access. If someone else already owns the group, you will not be connected to that group.

If you have auto connect enabled for the group, auto connect controls how you can disconnect from a group:

- You are not allowed to disconnect from the group. You have to first disable auto connect, and then disconnect from the group. The next time you start your computer, you will not be automatically connected to this group.

- You can disconnect from a USB device in the group, but note that the device will be connected again the next time you start your computer.

For this to work as expected, you should also choose to automatically start the **AnywhereUSB Manager** each time you start your computer.

For example, you can enable auto connect for a group that has a camera connected to a port in the group. Every time the computer starts, the **AnywhereUSB Manager** starts and automatically connects the camera to your computer.

Enable auto connect for a group

You can choose to automatically connect to a selected group each time you open the **AnywhereUSB Manager**.

Note You can disable auto connect at any time.

1. Open **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Right-click on the AnywhereUSB group to which you want to automatically connect.
5. Select **Enable Auto Connect**. If you were not already connected to the group, you are immediately connected to the group. A note appears next to the group name and in the Group Status pane to show that you are connected to the group.

Disable auto connect for a group

After you have enabled auto connect for a group, you can disable this option. You will no longer automatically connect to this group when you open the **AnywhereUSB Manager**.

1. Open the **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Expand a Hub to display the groups in the Hub.
4. Right-click on the AnywhereUSB group to which you no longer want to automatically connect at start up.
5. Select **Disable Auto Connect** to turn off the auto connect feature for the group.

Manage the list of known Hubs

You can create a list of Hubs to which your **AnywhereUSB Manager** is allowed to connect when you open it. The Hubs you add to the list can be on the same network as your computer, or on a different network.

Hubs that you have added to the known Hubs list display when you open the **AnywhereUSB Manager**. These Hubs are in addition to any Hubs that are automatically discovered if you have enabled the **Autofind Hubs feature**.

Add a Hub to the known Hub list

You can use one of two methods to manually add a Hub to the known Hubs list:

- Right-click method
- Known Hubs dialog

The Hubs can be on the same network as your computer, or on a different network.

Right-click Hub menu option

When you use this method, a duplicate connection for this Hub is made until you disable the Autofind Hubs feature in the **Preferences** dialog.

1. Open the **AnywhereUSB Manager**.
2. Right-click on a Hub name in the **AnywhereUSB Manager**. A short cut menu displays.
3. Click **Add to Known Hubs**. The Hub is added to the known hubs list.
4. To ensure that you don't have a duplicate connection for this Hub, you should navigate to **File** > **Preferences** and disable the Autofind Hubs feature.

(Optional) You can verify that the Hub was added to the list

1. Select the Hub and make a note of the IP address in the Hub status pane.
2. Select **Configure > Known Hubs**. The **Known Hubs** dialog appears.
3. Verify that the IP address for the Hub is in the list.

Known Hubs dialog

1. Open the **AnywhereUSB Manager**.
2. Select **Configure > Known Hubs**. The **Known Hubs** dialog appears.
3. Click **Add**. The **Add Known Hub** dialog appears.
4. In the **Hub Address** field, enter the Hub IP address or a network name, such as a DNS name, for the Hub.
5. If you want to update the TCP port number, click **Advanced**. The **Hub TCP port (most systems should leave at default)** field displays.
    a. In the **Hub TCP port (most systems should leave at default)** field, a TCP port number is entered by default. You can change this entry, but it is not recommended.
    b. Click **Standard** to hide the **Hub TCP port (most systems should leave at default)** field.
6. Click **OK**. The Hub appears in the Hub list in the **Known Hubs** dialog.
7. Click **Close** to close the **Known Hubs** dialog. The **AnywhereUSB Manager** attempts to connect to the new Hub.

Remove a Hub from the known Hub list

You can remove a known Hub that was added to the known Hubs list.

1. Open the **AnywhereUSB Manager**.
2. Select **Configure > Known Hubs**. The **Known Hubs** dialog appears.
3. From the list of known Hubs, select the Hub you want to remove.
4. Click **Remove**.
5. Click **Close** to close the **Known Hubs** dialog.

Working with the known Hubs list and the Autofind Hubs option

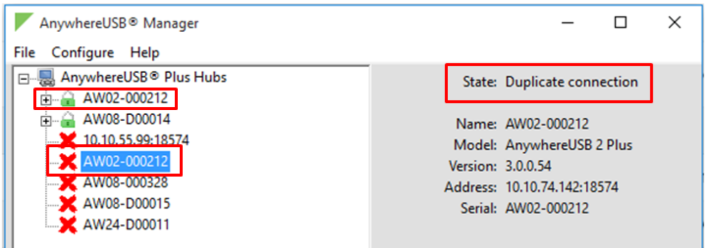
You should be aware of how the **Autofind Hubs** option works with the Hubs you add to the known Hubs list.

If you have the **Autofind Hubs option** selected for the Hub, when you open the **AnywhereUSB Manager**, all Hubs connected to the same network as your computer are automatically found and appear in the **AnywhereUSB Manager**. In addition, any Hubs you have added to the known Hubs list are found and also appear.

Duplicate connection

If you have added a Hub to the known Hub list that is on same network as your computer, and you have the **Autofind Hubs** feature enabled, the Hub is found twice and appears twice in the **AnywhereUSB Manager**. When you select the duplicate Hub in the **AnywhereUSB Manager**, the state for that Hub is noted as "Duplicate connection" in the **Status** field.

In this situation, the Hub added to the known Hubs list is considered a duplicate Hub, and should be removed from the known Hubs list.



AnywhereUSB® Plus Hubs
- AW02-000212
- AW08-D00014
- 10.10.55.99:18574
- AW02-000212
- AW08-000328
- AW08-D00015
- AW24-D00011

State: Duplicate connection

Name: AW02-000212
Model: AnywhereUSB 2 Plus
Version: 3.0.0.54
Address: 10.10.74.142:18574
Serial: AW02-000212

Considerations for removing a Hub on the same network as your computer

If you have the **Autofind Hubs** feature enabled and then remove a Hub from known Hubs list that was on the same network as your computer, the Hub will still be automatically found and connected to your computer when you open the **AnywhereUSB Manager**.

If you do not want the computer to be able to connect this Hub, you must de-select the **Autofind Hubs** option. Note, however, that if this option is de-selected, Hubs on the same network as your computer will not be automatically found. Only the Hubs in the list of known Hubs will be available when you open the **AnywhereUSB Manager**.

Note As an alternative, you can choose to hide a Hub that is automatically found. This ensures that while the Hub is still automatically found, it does not appear in the **AnywhereUSB Manager**.

Hide an individual Hub

You can choose to hide an individual Hub so that it does not appear in the **AnywhereUSB Manager**. For example, you can hide an unauthorized Hub, or a Hub which users shouldn't access.

- You can choose to hide Hubs that currently display in the **AnywhereUSB Manager**, such as an unauthorized Hub (which displays with a red X next to the Hub name), or a Hub which users shouldn't access. See Hide a Hub that displays in the AnywhereUSB Manager.
- You can also choose to hide Hubs that don't currently display in the **AnywhereUSB Manager**, but the client ID may have access in the future, such as a Hub on another network. See Hide a Hub that does not currently display in the AnywhereUSB Manager.

Note You can choose to automatically hide all unauthorized Hubs, which is a Hub that has failed to connect to your computer. See Hide all unauthorized Hubs.

Hide a Hub that displays in the AnywhereUSB Manager

Note After you have hidden a Hub, you can choose to re-display it. See Display a hidden Hub.

1. Open **AnywhereUSB Manager**.
2. Right-click on the Hub that you want to hide. The shortcut menu appears.
3. Click **Hide Hub**. The next time the **AnywhereUSB Manager** updates, the hidden Hub is removed from the Hub list and no longer displays.
4. You can display a hidden Hub when needed.

Hide a Hub that does not currently display in the AnywhereUSB Manager

Note After you have hidden a Hub, you can choose to re-display it. See Display a hidden Hub.

1. Open the **AnywhereUSB Manager**.
2. Select **Configure > Hidden Hubs**. The **Hidden Hubs** dialog appears.
3. Click **Add**. The **Add Hidden Hub** dialog appears.
4. In the **Hub Address** field, enter the Hub IP address.
5. If you want to update the TCP port number, click **Advanced**. The **Hub TCP port (most systems should leave at default)** field displays.
   a. In the **Hub TCP port (most systems should leave at default)** field, a TCP port number is entered by default. You can change this entry, but it is not recommended.
   b. Click **Standard** to hide the **Hub TCP port (most systems should leave at default)** field.
6. Click **OK**. The Hub appears in the Hub list in the **Hidden Hubs** dialog.
7. Click **Close** to close the **Hidden Hubs** dialog.

Display a hidden Hub

You can display any Hub that was hidden using the **Hide Hub** menu option.

1. Open **AnywhereUSB Manager**.
2. Choose **Configure > Hidden Hubs**. The **Hidden Hubs** dialog appears.
3. Click on the Hub that you no longer want to hide. To select more than one Hub, press CTRL as you select Hub.
4. Click **Remove**. The selected Hubs are removed from the list.
5. Click **Close**. The next time the **AnywhereUSB Manager** updates, the hidden Hubs appear in the list of Hubs.

Hide all unauthorized Hubs

You can choose to automatically hide all unauthorized Hubs, so they do not display in the **AnywhereUSB Manager**. An unauthorized Hub is a Hub that has failed to connect to your computer. A red X appears next to the Hub name.

- Automatically hide unauthorized Hubs
- Display unauthorized Hubs

Note You can choose to automatically hide any individual Hub. See Hide an individual Hub.

Automatically hide unauthorized Hubs

You can choose to automatically hide all unauthorized Hubs, which is a Hub that has failed to connect to your computer. An unauthorized Hub appears with a red X next to it in the list of Hubs in the **AnywhereUSB Manager**.

Note After you have hidden unauthorized Hubs, you can choose to re-display unauthorized, hidden Hubs. See Display unauthorized Hubs.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. Select the **Hide unauthorized Hubs** option.
4. Click **Save**. Hubs that have failed to connect no longer display in the **AnywhereUSB Manager**.

Display unauthorized Hubs

You can display the unauthorized Hubs that were hidden using the **Hide unauthorized Hubs** option.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. De-select the **Hide unauthorized Hubs** option.
4. Click **Save**. Hubs that have failed to connect now display in the **AnywhereUSB Manager**.

Use all Hub addresses

The AnywhereUSB Hub may have default IP addresses that are reported by mDNS to the **AnywhereUSB Manager**, but in many network environments, the **Manager** cannot connect to them. As part of normal operation, the **Manager** tries to sequentially connect to all of the Hub IP addresses, so if it starts trying these extra default IP addresses, it may take extra time (minutes) for the **Manager** to connect or reconnect.

You can use the **Use All Hub Addresses** option to determine whether the **AnywhereUSB Manager** is allowed to connect to extra default IP addresses. By default, this option is deselected and the **Manager** does not attempt to connect to these addresses.

Note This can also be done using a CLI command: USEALLHUBADDRS

1. Open **AnywhereUSB Manager**.
2. Choose **File** > **Preferences**. The **Preferences** dialog appears.

3. Determine your connection option:
   - **Not selected**: When **Use All Hub Addresses** is not selected, the **AnywhereUSB Manager** does not attempt to connect to the extra IP addresses. This is the default.
   - **Selected**: When **Use All Hub Addresses** is selected, the **AnywhereUSB Manager** attempts to connect to the extra IP addresses.
4. Click **Save** to save your change and close the dialog.

Minimize the AnywhereUSB Manager when launched

You can choose to automatically minimize the **AnywhereUSB Manager** when it launches.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. Click the **Setup** tab.
4. Determine whether you want to automatically minimize the **AnywhereUSB Manager** when it launches.
   - Select **Start Manager minimized** to automatically minimize the **AnywhereUSB Manager** when it launches.
   - De-select **Start Manager minimized** to open the **AnywhereUSB Manager** when it launches.
5. Click **Save**.

Autofind Hubs in the AnywhereUSB Manager

You can choose to automatically find Hubs connected to the network when **AnywhereUSB Manager** launches and repeatedly while the **AnywhereUSB Manager** is running, based on the interval specified in the **Preferences** dialog.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. Click the **Setup** tab.
4. Determine whether you want to automatically find Hubs on the network when **AnywhereUSB Manager** launches.
   - Select **Autofind Hubs** to automatically find Hubs on the network. This is the default.
   - De-select **Autofind Hubs** to ensure that the Hubs are not automatically found. In this case, you must manually add the Hubs to which you want to connect.
5. Click **Save**.

Specify search, response, and keepalive intervals for a Hub

You can specify the search and response time for Hubs on the network, and the keepalive intervals for the connection between the Hub and the **AnywhereUSB Manager**.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. Click the **Advanced** tab.

4. Enter the following:
   - **Search for Hubs every .... sec**: Specifies how often the **AnywhereUSB Manager** searches the local network to discover AnywhereUSB Hubs and refresh the **AnywhereUSB Manager** display.

     Note You cannot manually perform a refresh of the Hubs displayed in the **AnywhereUSB Manager**.

   - **Wait for Hub response for .... sec**: Specifies the time interval from the last discovery refresh that the **AnywhereUSB Manager** will stop looking for more Hubs.
   - **Send Keep-Alive every ... sec**: Specifies how often the **AnywhereUSB Manager** sends a keepalive request to the Hubs connected to the network. This impacts network utilization because each **AnywhereUSB Manager** will send one packet at this interval to each Hub to which it is connected.
   - **Keep-Alive Timeout ... sec**: Specifies how long the **AnywhereUSB Manager** should wait for a keepalive response. When the value of the response time is reached, the Manager decides that a Hub is no longer available, and the computer is disconnected from all groups and devices on that Hub.
     - The keepalive timeout value would need to be longer if the network has more latency (such as a cellular or satellite link), or an internet link with unreliable packet delivery.
     - If the value is too short, devices will be disconnected, which may have an adverse affect on some devices, such as USB memory.
     - If the value is too long, Hubs that are removed from the network will not be noticed as gone for a long time, and devices that are no longer connected will be unresponsive for a long time.
5. Click **Save**.

Cycle the power to a device connected to the Hub

This feature enables you to power cycle one selected USB device that is connected directly to the AnywhereUSB Hub or to a downstream USB hub. This resets the USB device and has the same effect as removing the USB device from the Hub and then reconnecting it.

Note If an externally powered USB device (one that is not powered by the Hub) is connected to the Hub, the power cycle feature may have no effect on the USB device.

The USB device you choose to power cycle must be assigned to a group that you are allowed to access. When you use this feature, the power supplied by the port to the USB device is turned off and then turned on. The USB device remains connected to the group and the Hub during the process.

1. Open **AnywhereUSB Manager**.
2. Expand the Hub and group to which the USB device is connected.
3. Right-click on the USB device and click **Power Cycle Device**. The power supplied to the port to the USB device is turned off and then turned on.

Manage Hub credentials

You can manually add, update, or remove the certificate associated with a Hub on the **AnywhereUSB Manager**. The Hub and the **AnywhereUSB Manager** must have matching certificates to be able to communicate.

This feature works with the **Auto-register Hub Cert** option in the **Preferences** dialog. You should determine whether you want to automatically register a Hub's certificate with the **AnywhereUSB Manager**.

Enable and disable the auto-register Hub certificate

You can choose to automatically find Hubs connected to the network when **AnywhereUSB Manager** launches.

1. Open **AnywhereUSB Manager**.
2. Choose **File > Preferences**. The **Preferences** dialog appears.
3. Click the **Setup** tab.
4. Determine whether you want to automatically register a Hub with the **AnywhereUSB Manager**.
    - Select **Auto-register Hub Cert** to automatically register a Hub certificate with the **AnywhereUSB Manager**. This is the default.
    - De-select **Auto-register Hub Cert** to ensure that the Hub certificates are not automatically registered. In this case, you must manually register a certificate for each Hub.
5. Click **Save**.

Update a Hub certificate

You can choose to manually update a Hub's certificate and register a new certificate with the **AnywhereUSB Manager**. This ensures that the Hub and the **AnywhereUSB Manager** can communicate.

1. Open **AnywhereUSB Manager**.
2. Choose **Configure** > **Manage Hub Credentials**. The **Manage Hub Credentials** dialog appears.
3. Select the Hub for which you want to update the certificate.
4. Click **Update**. The **Choose a credential file** window appears.
5. Browse for the new certificate file and click **Open**. The file should have a .pem extension.
6. An update message displays in the **Manage Hub Credentials** dialog.
7. Click **Close**.

Remove a Hub certificate

You can choose to remove a Hub to which you no longer want the **AnyhwereUSB Manager** to connect.

After you have removed a Hub certificate, if you have the Auto-register Hub Cert option selected in the **Preferences** dialog, a new certificate for the Hub is automatically registered with the **AnyhwereUSB Manager** at the next connection attempt or the next time the Hub is discovered by the Manager. If this option is not selected and you want the **AnywhereUSB Manager** to connect to the Hub, you can manually add the Hub to register a new certificate.

1. Open **AnywhereUSB Manager**.
2. Choose **Configure** > **Manage Hub Credentials**. The **Manage Hub Credentials** dialog appears.
3. Select the Hub that you want to remove.
4. Click **Close**.

Add a Hub certificate

You can manually add a Hub, which registers the Hub's certificate with the **AnywhereUSB Manager**.

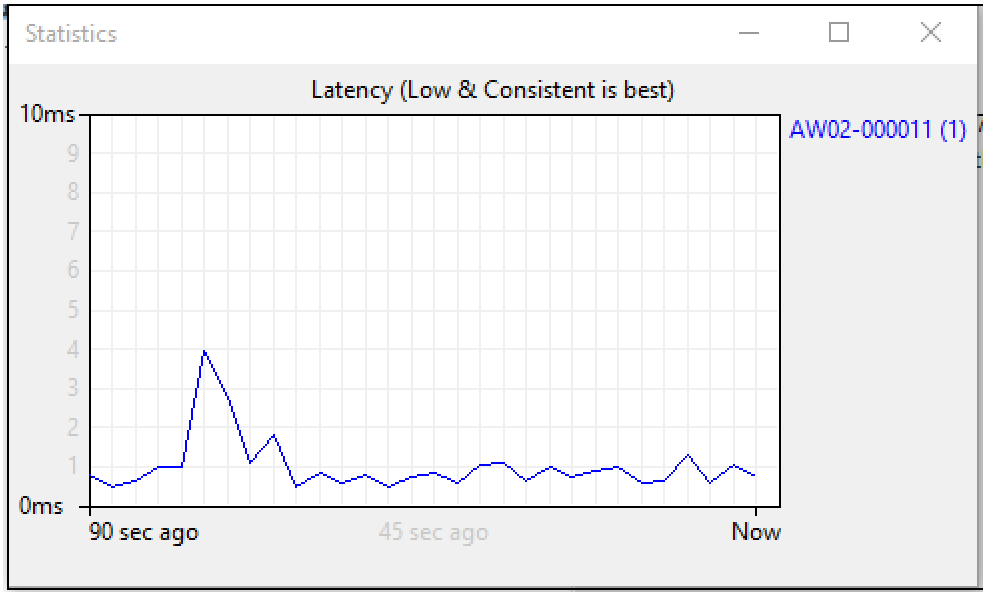
1. Open **AnywhereUSB Manager**.
2. Choose **Configure** > **Manage Hub Credentials**. The **Manage Hub Credentials** dialog appears.
3. In the **Serial number** field, enter the Hub's serial number.
4. Click **Add**. The **Choose a credential file** window appears.
5. Browse for the new certificate file and click **Open**. The file should have a .pem extension.
6. An update message displays in the **Manage Hub Credentials** dialog.
7. Click **Close**.

View latency graph

You can review the relative latency of all of the Hubs connected to the network.

Note The**Latency Graph**menu item is not available when the**AnywhereUSB Manager**is installed in service mode.

1. Open the **AnywhereUSB Manager**.
2. Select **Help > Latency graph** to display the latency graph.



Create support log file

You can use the **Create Support File** feature in the **AnywhereUSB Manager** when you need to collect logs and other information for Digi Technical Support. The information is saved to a .bin file which you can send to technical support.

The location in which the file is saved depends on whether the **Manager** was installed in service or stand-alone mode. After you have created the file, a dialog displays the location in which the .bin file was saved.

The file is overwritten each time you create a file. If you want to save a file before it is overwritten, rename the file or move it to a different location.

1. Open **AnywhereUSB Manager**.
2. Choose **File** > **Preferences**. The **Preferences** dialog appears.
3. Choose **Help** > **Create Support File**. The support file is created. When complete, a dialog displays, showing you the location of the file.
4. Make a note of the file location.
5. Click **OK** to close the dialog.
6. Navigate to the file location and copy it. You can then email the copy to Digi Technical Support.

Note If you installed the **AnywhereUSB Manager** in service mode, you must have Administrator rights on the computer to copy the file.

Always display the AnywhereUSB Manager on top

You can choose to always display the **AnywhereUSB Manager** on top of all open windows. This feature is disabled by default.

1. Open the **AnywhereUSB Manager**.
2. Select **Help > Always on top**. This option toggles between disabled and enabled, and is disabled by default. When it is enabled, a check mark displays next to the option.

View the AnywhereUSB Manager system messages

You can view the system message log of the **AnywhereUSB Manager** events. The date and time at which an event occurred is listed, as well as the event type and additional information. A new log is created each time you start the **AnywhereUSB Manager**.

The system message log is used for troubleshooting.

1. Open the **AnywhereUSB Manager**.
2. Select **Help > System Messages**. The **System Messages** dialog appears.
   - Click **Refresh** to update the system messages.
   - Click **Clear Log** to clear the system messages from the log.
   - Click **Copy to Clipboard** to copy the messages to the Windows clipboard. You can then paste the messages into another application or document.
3. Click **Close** to close the **System Messages** dialog.

View AnywhereUSB version and license information

You can view version and license information about the AnywhereUSB Hub.

The version numbers for the currently installed version of the **AnywhereUSB Manager**, the driver, and the installer are listed at the top of the screen.

1. Open the **AnywhereUSB Manager**.
2. Select **Help > About**. The **License** dialog appears.
3. View the version numbers at the top of the screen.
   - **Manager Version**: The currently installed version of the **AnywhereUSB Manager**.
   - **Driver Version**: The version of the Windows driver installed when the **Manager** was installed.
   - **Installer Version**: The version of the AnywhereUSB installer that was used to install the **Manager** and the Windows driver.
4. In the **License** window, scroll down to review the license information.
5. Click **Close** to close the dialog.

Restore AnywhereUSB Manager default configuration

You can restore the **AnywhereUSB Manager** to the default settings. During this process, you have the option to keep your currently configured client ID and credentials during this process. See Client ID for more information about how the client ID is used by your computer and the Hub to create a connection.

- Keep the current client ID
- Change the client ID

Keep the current client ID

To restore the Hub's default settings and keep your currently configured client ID and identity certificate:

1. Open the **AnywhereUSB Manager**.
2. Select **File > Preferences**. The **Preferences** dialog appears.
3. Click the **Setup** tab.
4. Click **Restore default settings**. A dialog appears.
5. Select the **Keep Client ID** option. This is selected by default.
6. Click **OK**. The **AnywhereUSB Manager** closes automatically. The next time you launch the **AnywhereUSB Manager**, the default settings will be restored.

Change the client ID

To restore the Hub's default settings and change your currently configured client ID and credentials:

1. Open the **AnywhereUSB Manager**.
2. Select **File > Preferences**. The **Preferences** dialog appears.
3. Click **Restore default settings**. A dialog appears.
4. De-select the **Keep Client ID** option.

5. Click **OK**. The **AnywhereUSB Manager** closes automatically.
6. Open the **AnywhereUSB Manager** again. The **Client ID** confirmation dialog appears.
7. Enter a new, unique client ID.
8. Click **OK**. The **AnywhereUSB Manager** launches.

Access the online help from the AnywhereUSB Manager

1. Open the **AnywhereUSB Manager**.
2. Click **Help > Online Manual** to launch the online help file.

AnywhereUSB Manager window

The **AnywhereUSB Manager** displays AnywhereUSB Hubs, groups, and USB devices. Click the plus sign next to each name in the window to display a hierarchy of found Hubs, groups, and USB devices.

Information about the icons and messages in the screen and the menu options can be found here:

- AnywhereUSB Manager icons and toolbar
- AnywhereUSB Manager menu options
- AnywhereUSB Manager Hub connection messages

You can use the menus associated with the Hubs, groups, and USB devices to configure local names, preferences, and connections. Right-click on a Hub, group, or device name to display the menus.

- AnywhereUSB Manager Hub menu options
- AnywhereUSB Manager group menu options
- AnywhereUSB Manager USB device menu options

Click on a Hub, group, or device name to display information about the selected Hub, group, or device in the status pane on the right side of the **AnywhereUSB Manager**.

- AnywhereUSB Manager Status pane
- AnywhereUSB Manager Hub Status pane
- AnywhereUSB Manager Group Status pane
- AnywhereUSB Manager Device Status pane

AnywhereUSB Manager icons and toolbar

This section explains how to use the icons in the **AnywhereUSB Manager** and what they represent.

The icons in the **AnywhereUSB Manager** show the status of a Hub or a USB device.

Icon	Location	Description
🔒	Hub	Green lock: Active and secure connection between the Hub and the PC.

Icon	Location	Description
●	Hub	Yellow dot: The PC and Hub are attempting to connect.
✖	Hub	Red X: Connection between the Hub and the PC failed.
❓	USB device	Question mark: Signifies unknown device class.

The toolbar icons manage the **AnywhereUSB Manager** dialog.

Icon	Description
—	Minimizes the **AnywhereUSB Manager** into the task bar and the notification area of the task bar.
□	Maximizes the **AnywhereUSB Manager**.
×	Minimizes the **AnywhereUSB Manager** into the notification area of the task bar.

AnywhereUSB Manager menu options

You can use the menu options to view AnywhereUSB Hub information.

- **File** > **Refresh**: Select **File** > **Refresh** to refresh the Hub information.
- File > Preferences
- File > Exit
- Configure > Known Hubs
- Configure > Hidden Hubs
- Configure > Manage Hub Credentials
- Help > System Messages
- Help > Latency graph
- Help > Always on Top
- Help > Create Support File
- Help > Online Manual
- Help > About

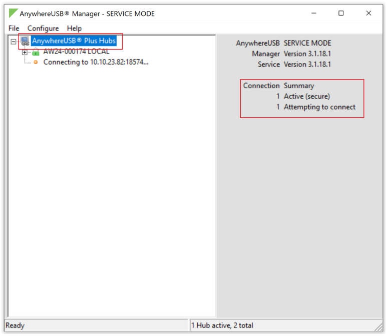
AnywhereUSB Manager Hub connection messages

The number of Hubs connected to the AnywhereUSB Manager can be reviewed in messages on the screen.

Connection summary

A summary of the connection status for each of the Hubs listed in the **AnywhereUSB Manager**. Click on the top node in the **Manager** to display connection status information in the **AnywhereUSB Manager** status pane.

- **Active (secure)**: The number of Hubs that are currently connected to the **AnywhereUSB Manager**.
- **Attempting to connect**: The number of Hubs attempting to connect to the **AnywhereUSB Manager**.
- **Unable to connect**: The number of Hubs that are unable to connect to the **AnywhereUSB Manager**.



Connecting to 10.10.23.82:18574...

AnywhereUSB SERVICE MODE
Manager Version 3.1.18.1
Service Version 3.1.18.1

Connection Summary
1 Active (secure)
1 Attempting to connect

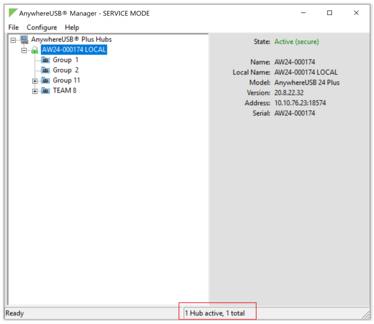
Ready | 1 Hub active, 2 total

For more information about this screen, see AnywhereUSB Manager Status pane.

Hub message

The number of total and active Hubs displays in a message at the bottom of the screen.

- **Active**: The number of Hubs that are connected to the **AnywhereUSB Manager**.
- **Total**: The total number of unhidden hubs available to the **Manager**. Hidden Hubs are not included in this count. For information about hidden hubs, see Hide an individual Hub and Hide all unauthorized Hubs.



State: Active (secure)

Name: AW24-000174
Local Name: AW24-000174 LOCAL
Model: AnywhereUSB 24 Plus
Version: 20.8.22.32
Address: 10.10.76.23:18574
Serial: AW24-000174

Ready | 1 Hub active, 1 total

AnywhereUSB Manager Hub menu options

Right-click on a Hub name in the **AnywhereUSB Manager** to configure and maintain the Hub.

- Open Web UI
- Assign Local Name

- Add to Known Hubs
- Hide Hub

AnywhereUSB Manager group menu options

Right-click on a group name in the **AnywhereUSB Manager** to configure and maintain the group.

- Connect to Group
- Disconnect from Group
- Enable Auto Connect
- Disable Auto Connect
- Assign Local Name

AnywhereUSB Manager USB device menu options

Right-click on a USB device name in the **AnywhereUSB Manager** to configure and connect to the USB device.

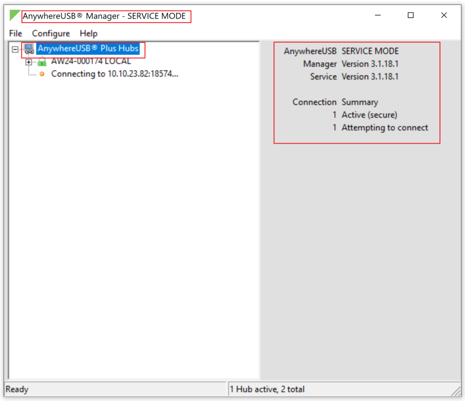
- Connect to Device
- Connect to Group
- Disconnect from Device
- Power Cycle Device
- Assign Local Name

AnywhereUSB Manager Status pane

When you select the top node the **AnywhereUSB Manager**, information about the **Manager** displays in the Manager Status pane. The information displayed depends on whether the **Manager** was installed in service mode or stand-alone mode.

Service mode

When installed in service mode, the **AnywhereUSB Manager** dialog title is "AnywhereUSB SERVICE MODE".



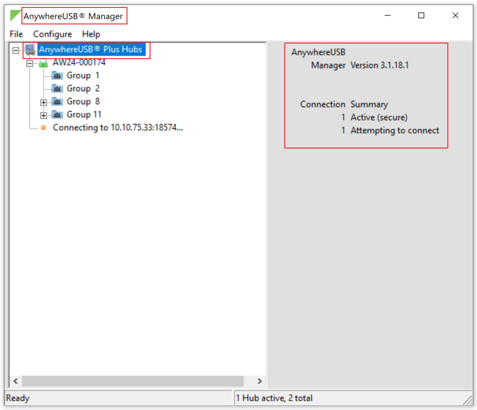
Connecting to 10.10.23.82:18574...

```
AnywhereUSB  SERVICE MODE
    Manager  Version 3.1.18.1
    Service  Version 3.1.18.1

 Connection  Summary
          1  Active (secure)
          1  Attempting to connect
```

Ready | 1 Hub active, 2 total

Stand-alone mode



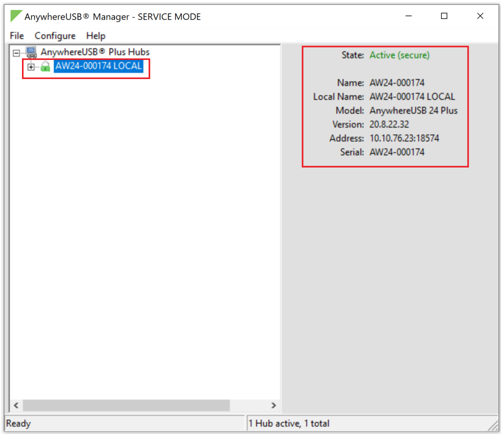
```
AnywhereUSB
    Manager  Version 3.1.18.1

 Connection  Summary
          1  Active (secure)
          1  Attempting to connect
```

Label	Description
Mode	The **AnywhereUSB Manager** mode that was selected during installation: stand-alone or service mode. See Step 2: Determine how to run AnywhereUSB Manager: Service or stand-alone. ■ Service mode: AnywhereUSB SERVICE MODE ■ Stand-alone mode: AnywhereUSB
Manager Version	The version number of the currently installed version of the **AnywhereUSB Manager**.
Service Version	The version number of the currently running service. Note This displays only when the **Manager** is installed in service mode.
Connection Summary	A summary of the connection status for each of the Hubs listed in the **AnywhereUSB Manager**. Click on the top node in the **Manager** to display connection status information in the **AnywhereUSB Manager** status pane. ■ **Active (secure)**: The number of Hubs that are currently connected to the **AnywhereUSB Manager**. ■ **Attempting to connect**: The number of Hubs attempting to connect to the **AnywhereUSB Manager**. ■ **Unable to connect**: The number of Hubs that are unable to connect to the **AnywhereUSB Manager**.

AnywhereUSB Manager Hub Status pane

When you select an AnywhereUSB Hub in the **AnywhereUSB Manager**, information about the Hub displays in the Hub Status pane.

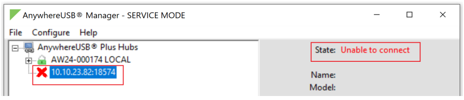


State: Active (secure)

Name: AW24-000174
Local Name: AW24-000174 LOCAL
Model: AnywhereUSB 24 Plus
Version: 20.8.22.32
Address: 10.10.76.23:18574
Serial: AW24-000174

Ready

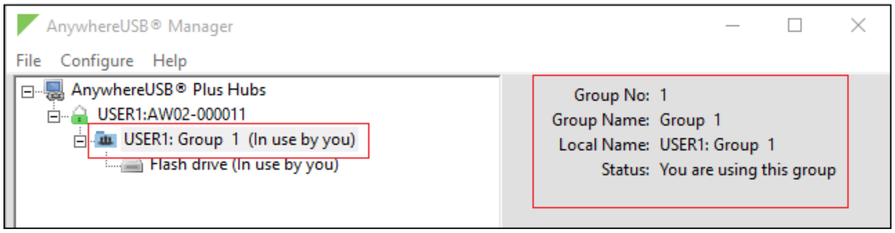
1 Hub active, 1 total

Label	Description
State	The current state of the Hub. Options are: ■ **Connecting** ■ **Authenticating** ■ **Active (secure)**: The **AnywhereUSB Manager** has connected to the Hub. A green lock icon appears next to the Hub name. The message **Active (secure)** displays in green. ■ **Unable to connect**: The **AnywhereUSB Manager** was not able to connect to the Hub. The message **Unable to connect** displays in red.  AnywhereUSB® Manager - SERVICE MODE File Configure Help AnywhereUSB® Plus Hubs AW24-000174 LOCAL 10.10.23.82:18574 State: Unable to connect Name: Model: ■ **Attempting to connect**: Displays when the **AnywhereUSB Manager** is trying to connect to the Hub but a connection has not yet been made. ■ **Error**: If an error has occurred, a red X icon appears next to the Hub name. ■ **Duplicate connection**: If a Hub that is on the same network as your computer has been added to the known Hubs list, and the **Autofind Hubs** feature is enabled, a duplicate Hub displays in the AnywhereUSB Manager. The duplicate Hub will be in the **Duplicate connection** state. See Manage the list of known Hubs. ■ **Unregistered client ID**: The client ID is not in the client list for the Hub. The Hub administrator needs to allow each new client ID by adding the client ID to the client list. See AnywhereUSB Manager icons and toolbar for more information about the Hub icons.
Name	The name of the Hub supplied by the Hub. The default value for the Hub name is the serial number assigned to the Hub. You can change the Hub name in the **Ethernet Network** section of the web UI. See Rename the Hub.
Local Name	A descriptive local name for the Hub. The local name also displays in the tree view in the left-hand pane in the **AnywhereUSB Manager**. The local name is local to the computer on which the **AnywhereUSB Manager** is running. You can change the local name using the Assign Local Name menu option for the Hub.
Model	The model name for the AnywhereUSB Hub.

Label	Description
Version	The version number of the firmware running on the Hub.
Address	The network address of the Hub.
Serial	The serial number of the Hub, which is found on the Hub label.

AnywhereUSB Manager Group Status pane

When you select a group in the **AnywhereUSB Manager**, information about the group displays in the Group Status pane.

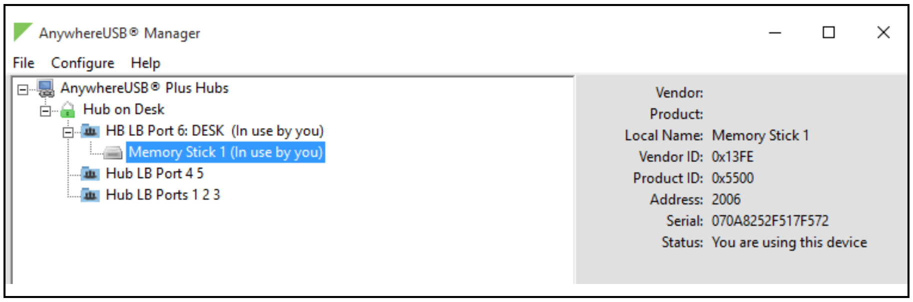


Group No: 1
Group Name: Group 1
Local Name: USER1: Group 1
Status: You are using this group

Label	Description
Group No	The group number from the Hub.
Group Name	The name of the group supplied by the Hub. By default, a group is named "Group" appended by a consecutive number, such as Group 1, Group 2, and so on. You can change the group name in the **AnywhereUSB** screen in the web UI. See Create groups and assign ports to the group.
Local Name	A descriptive local name for the group. The local name also displays in the tree view in the left-hand pane in the **AnywhereUSB Manager**. The local name is local to the computer on which the **AnywhereUSB Manager** is running. You can change the local name using the Assign Local Name menu option for the group.
Status	A status message indicates whether a user is currently connected this group. Options are: ■ You are using this group ■ No one is using this group ■ In use by <*client ID*> at <*machine name*>

AnywhereUSB Manager Device Status pane

When you select a USB device in a group in the **AnywhereUSB Manager**, information about the device displays in the Device Status pane.



AnywhereUSB® Plus Hubs
Hub on Desk
HB LB Port 6: DESK (In use by you)
Memory Stick 1 (In use by you)
Hub LB Port 4 5
Hub LB Ports 1 2 3

Vendor:
Product:
Local Name: Memory Stick 1
Vendor ID: 0x13FE
Product ID: 0x5500
Address: 2006
Serial: 070A8252F517F572
Status: You are using this device

Label	Description
Vendor	Name of the USB device vendor, if supplied by the device.
Product	Name of the USB product, if supplied by the device.
Local Name	A descriptive local name for the USB device. The local name also displays in the tree view in the left-hand pane in the **AnywhereUSB Manager**. The local name is local to the computer on which the **AnywhereUSB Manager** is running. You can change the local name using the **Assign a Local Name** menu option for the device. See Assign a local name to a USB device.
Vendor ID	The USB vendor ID.
Product ID	The USB product ID.
Address	The USB device address that helps to identify a device.
Serial	The serial number of the USB device, if supplied by the device.
Status	A status message indicates whether a user is currently using this device. Options are: ■ You are using this device ■ No one is using this device ■ In use by <*client ID*> at <*machine name*> ❓ A question mark icon displays if the device class is unknown.

Set Hub preferences

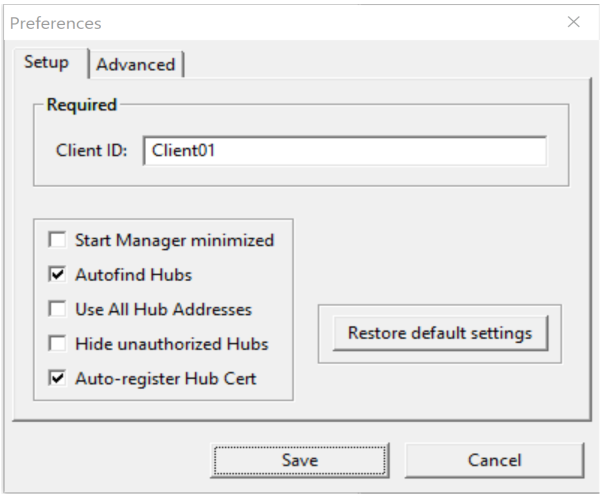
In the **AnywhereUSB Manager**, you can set preferences for keepalive time messages and responses and how often the **AnywhereUSB Manager** searches for a Hub and the Hub response time.

Click **File > Preferences** to display the **Preferences** dialog.

Setup tab

- Client ID
- Start Manager minimized

- Autofind Hubs
- Use All Hub Addresses
- Hide unauthorized Hubs
- Auto-register Hub Cert
- Restore default settings


Preferences

Setup | Advanced

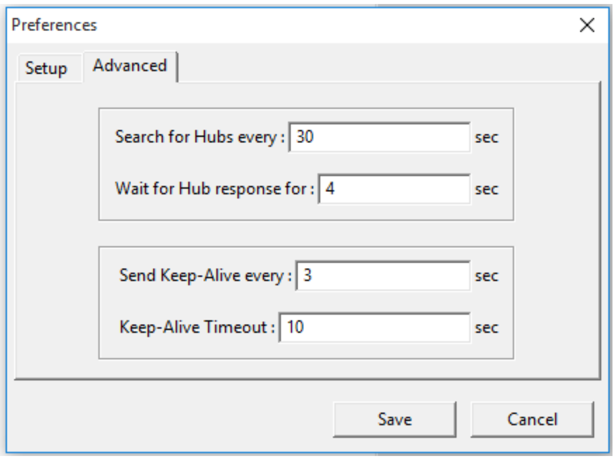
Required

Client ID: Client01

□ Start Manager minimized
☑ Autofind Hubs
□ Use All Hub Addresses
□ Hide unauthorized Hubs
☑ Auto-register Hub Cert

Advanced tab

Specify search, response, and keepalive intervals for a Hub


Preferences

Setup | Advanced

Search for Hubs every : 30 sec

Wait for Hub response for : 4 sec

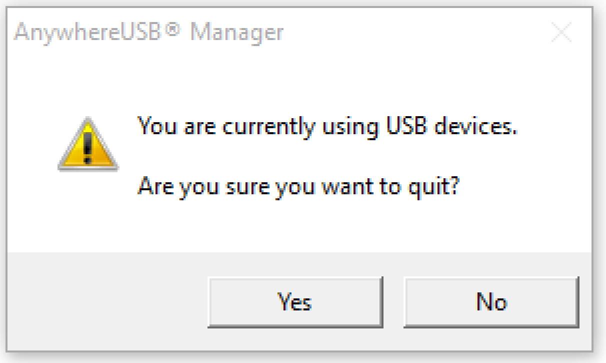
Send Keep-Alive every : 3 sec

Keep-Alive Timeout : 10 sec

Exit the AnywhereUSB Manager

You can log out of the AnywhereUSB Manager close the dialog.

1. Open the **AnywhereUSB Manager**.
2. Click **File > Exit** to disconnect all USB devices connected to your computer, close all connections, and close the **AnywhereUSB Manager**.
3. If you are connected to any USB devices, a confirmation dialog appears.



AnywhereUSB® Manager

You are currently using USB devices.

Are you sure you want to quit?

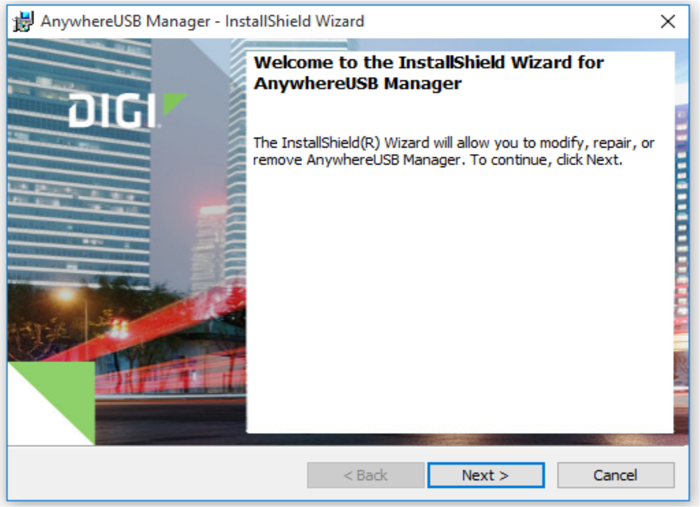
4. Click **Yes** to exit the **AnywhereUSB Manager**.

Uninstall the AnywhereUSB Manager

You can uninstall the **AnywhereUSB Manager** if needed.

Note You can also uninstall the **AnywhereUSB Manager** from the Windows Control Panel.

1. Locate the **AnywhereUSB Manager** installer. You must run the same version of the installer to uninstall the **AnywhereUSB Manager** that you used to install it.
   - If you saved the installer when you originally installed the **AnywhereUSB Manager**, navigate to that location on your computer.
   - If you did not, you can download the installer from the Support Tools website.
     a. Navigate to https://www.digi.com/support#support-tools.
     b. From the **Support Downloads** section, click **Drivers**.
     c. Find and select **AnywhereUSB Plus** from the product list.
     d. Select your **AnywhereUSB Plus** model.
     e. Select and download the appropriate software for your operating system.
2. Click on the downloaded software to launch the **AnywhereUSB Manager** installation wizard. The **Welcome** screen appears.

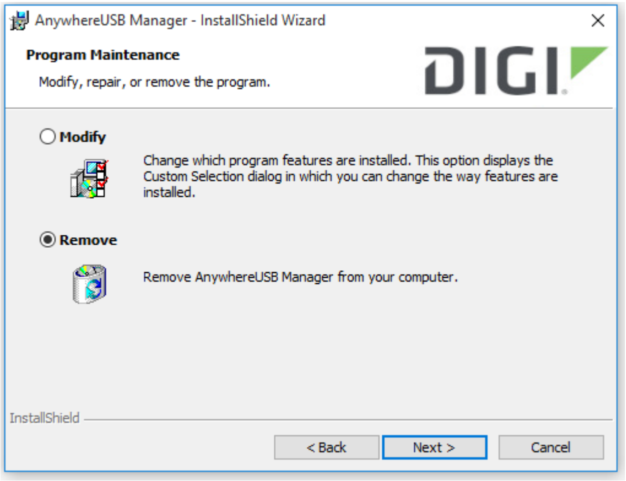


AnywhereUSB Manager - InstallShield Wizard

Welcome to the InstallShield Wizard for AnywhereUSB Manager

The InstallShield(R) Wizard will allow you to modify, repair, or remove AnywhereUSB Manager. To continue, click Next.

3. Click **Next**. The **Program Maintenance** screen appears.
4. Select **Remove**.



AnywhereUSB Manager - InstallShield Wizard

Program Maintenance

Modify, repair, or remove the program.

○ **Modify**

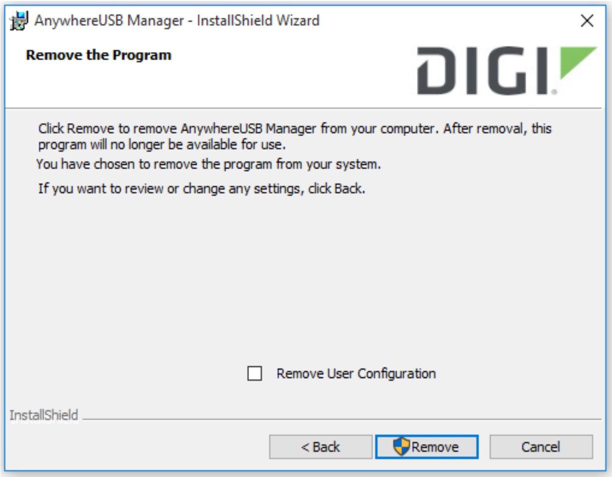
Change which program features are installed. This option displays the Custom Selection dialog in which you can change the way features are installed.

● **Remove**

Remove AnywhereUSB Manager from your computer.

5. Click **Next**. The **Remove the Program** screen appears.
6. Determine whether you want to remove the AnywhereUSB configuration settings that you have selected.
   - Do not select **Remove User Configuration**: The configuration settings you have made are retained and re-applied the next time you install the **AnywhereUSB Manger**. This is the default.
   - Select **Remove User Configuration**: The configuration settings you have made are not

retained and removed with the program.



AnywhereUSB Manager - InstallShield Wizard

Remove the Program

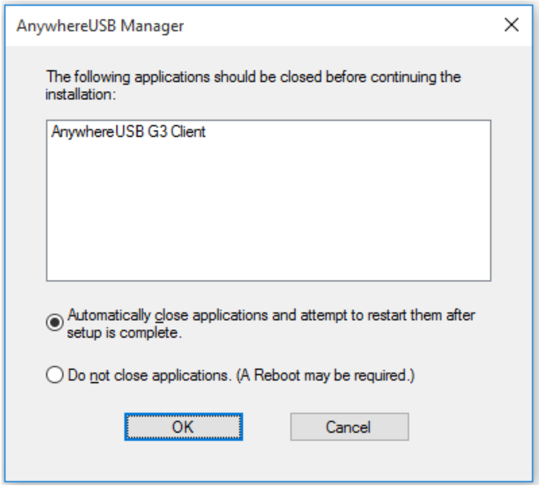
Click Remove to remove AnywhereUSB Manager from your computer. After removal, this program will no longer be available for use.

You have chosen to remove the program from your system.

If you want to review or change any settings, click Back.

☐ Remove User Configuration

7. Click **Remove**. If the **AnywhereUSB Manager** is open, the following dialog displays. Do not change the default settings.



AnywhereUSB Manager

The following applications should be closed before continuing the installation:

AnywhereUSB G3 Client

● Automatically close applications and attempt to restart them after setup is complete.

○ Do not close applications. (A Reboot may be required.)

8. Click **OK**. A progress bar appears.
9. When the uninstall is complete, the **InstallShield Wizard Completed** screen appears.
10. Click **Finish** to complete the uninstall and close the dialog.

Configuration and management

This chapter contains the following topics:

Review AnywhereUSB Plus default settings

You can review the default settings for your AnywhereUSB Plus device by using the local WebUI or Digi Remote Manager:

Local WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access. See Open the web user interface for details.
2. On the menu, click **System** > **Device Configuration**.

Digi Remote Manager

1. If you have not already done so, connect to your Digi Remote Manager account.
2. Click **Device Management** to display a list of your devices.
3. Locate and select your device as described in Use Digi Remote Manager to view and manage your device.
4. Click **Configure**.

The following tables list important factory default settings for the AnywhereUSB Plus.

Default interface configuration

Interface type	Preconfigured interfaces	Devices	Default configuration
Wide Area Networks (WANs) (Available only on the AnywhereUSB Plus 8 and 24 models.)	■ Modem	■ WWAN1 cellular modem	■ Firewall zone: External ■ WAN priority: Metric=3 ■ SIM failover after 5 attempts
Ethernet Network	■ ETH1	■ Ethernet: ETH1	■ Firewall zone: Edge ■ DHCP client enabled
	■ Loopback	■ Ethernet: Loopback	■ Firewall zone: Loopback ■ IP address: 127.0.0.1/8
	■ Default IP	■ Ethernet: ETH1	■ Firewall zone: Setup ■ IP address 192.168.210.1/24
	■ Default Link-local IP	■ Ethernet: ETH1	■ Firewall zone: Setup ■ IP address 169.254.100.100/16
	■ ETH2 (Available on the AnywhereUSB Plus 24 model only.)	■ Ethernet: ETH2	■ Firewall zone: Edge ■ DHCP client enabled

Other default configuration settings

Feature	Configuration
Central management	■ Digi Remote Manager enabled as the central management service.
Security policies	■ Packet filtering allows all outbound traffic. ■ SSH and web administration: • Enabled for local administration • Firewall zone: Set up
Monitoring	■ Device heath metrics uploaded to Digi Remote Manager at 60 minute interval. ■ SNMP: Disabled

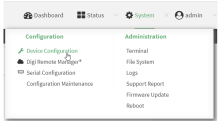
Change the default password for the admin user

The unique, factory-assigned password for the default **admin** user account is printed on the bottom label of the device and on the loose label included in the package. When you first log into the WebUI or the command line, you will be required to change the password for the **admin** user prior to being able to save any configuration changes.

If you erase the device configuration or reset the device to factory defaults, the password for the **admin** user will revert to the original, factory-assigned default password.

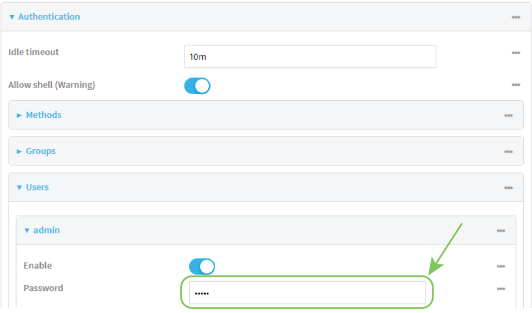
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Authentication** > **Users** > **admin**.
4. Enter a new password for the admin user. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.



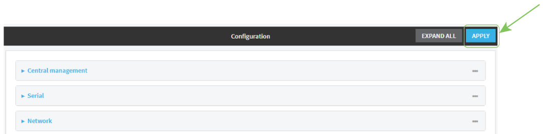
Idle timeout 10m

Allow shell (Warning)

Enable

Password •••••

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Set a new password for the admin user. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.

   ```
   (config)> auth user admin password new-password
   (config)>
   ```
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configuration methods

There are three methods for configuring your AnywhereUSB Plus device:

- Web interface

  The local web interface on the Hub, which includes a separate page for all AnywhereUSB Plus configuration.

  - See Open the web user interface for information about accessing the web interface from a Hub.
  - See Configure the AnywhereUSB in the web user interface for more information about using the local web interface to manage and configure your AnywhereUSB Plus device.

- Central management

  Central management using the Digi Remote Manager, a cloud-based device management and data enablement platform that allows you to connect any device to any application, anywhere. See Using Digi Remote Manager for more information about using the Remote Manager to manage and configure your AnywhereUSB Plus device.

- Command line

  A robust command line allows you to perform all configuration and management tasks from within a command shell. Both the Remote Manager and the local web interface also have the option to open a terminal emulator for executing commands on your AnywhereUSB Plus device. See Using the command line for more information about using the command line to manage and configure your AnywhereUSB Plus device.

In this guide, task topics show how to perform tasks:

☰ **WebUI**

Shows how to perform a task by using the local web interface.

⌨ **Command line**

Shows how to perform a task by using the command line interface.

Open the web user interface

You can open the web user interface for a selected AnywhereUSB Hub from the **AnywhereUSB Manager**. The information in the web UI is unique for each Hub, so make sure you select the desired Hub before you open the web UI.

By default, the web UI **Dashboard** appears when you open the web UI for a Hub, and displays current Hub status information.

Initial launch of the web UI and logging in

The first time you launch the web UI, a warning dialog may appear if your internet connection is not private. In this situation, continue to access the device, and a log in dialog appears. If your internet connection is private, only the log in dialog appears. The user name is **admin** and the default password is located on the label on the bottom of the Hub. Note that the password is case-sensitive and must be typed in exactly as it appears on the label.

After the initial launch of the web UI

Make sure that you have changed the default password on the Hub for the default **admin** user. This ensures that when you log into the WebUI or the command line you are able to save any configuration changes you have made. See Change the default password for the admin user.

WARNING! If you do not replace the default password on the Hub for the default admin user, you are able to stage configuration changes, but **you will not be able to save the configuration changes**.

1. Open the **AnywhereUSB Manager**.
2. Expand **AnywhereUSB Hubs** to display the Hubs.
3. Right-click on the Hub that you want to configure or maintain.
4. Click **Open Web UI**. The web UI **Dashboard** displays by default.
5. Change the default password on the Hub for the default **admin** user if you have not already done so. See Change the default password for the admin user.

   WARNING! If you do not replace the default password on the Hub for the default admin user, you are able to stage configuration changes, but **you will not be able to save the configuration changes**.

6. When you are done working in the web UI, close the browser window.

Using Digi Remote Manager

By default, your AnywhereUSB Plus device is configured to use Digi Remote Manager as its central management server. No configuration changes are required to begin using the Remote Manager.

For information about configuring central management for your AnywhereUSB Plus device, see Central management .

Access Digi Remote Manager

To access Digi Remote Manager:

1. If you have not already done so, go to https://myaccount.digi.com/ to sign up for a Digi Remote Manager account.

   Check your email for Digi Remote Manager login instructions.
2. Go to remotemanager.digi.com.
1. Enter your username and password.

   The Digi Remote Manager Dashboard appears.

Using the command line

The Digi AnywhereUSB Plus device provides a command-line interface that you can use to configure the device, display status and statistics, update firmware, and manage device files.

See Command line interface for detailed instructions on using the command line interface and see Command line reference for information on available commands.

Access the command line interface

You can access the AnywhereUSB Plus command line interface using an SSH connection or a serial connection. You can use an open-source terminal software, such as PuTTY or TeraTerm, to access the device through one of these mechanisms.

You can also access the command line interface in the WebUI by using the **Terminal**, or the Digi Remote Manager by using the **Console**.

To access the command line, your device must be configured to allow access, and you must log in as a user who has been configured for the appropriate access. For further information about configuring access to these services, see:

- WebUI: Configure the web administration service
- SSH: Configure SSH access

Log in to the command line interface

Command line

1. Connect to the AnywhereUSB Plus device by using a serial connection, SSH, or the **Terminal** in the WebUI or the **Console** in the Digi Remote Manager. See Access the command line interface for more information.
    - For serial connections, the default configuration is:
        - **115200** baud rate
        - **8** data bits
        - **no** parity
        - **1** stop bit
        - **no** flow control
    - For SSH connections, enter the device's default IP address.
2. At the login prompt, enter the username and password of a user with Admin access:

    ```
    login: admin
    Password: **********
    ```

    The default username is **admin**. The default unique password for your device is printed on the device label.
3. Depending on the device configuration, you may be presented with another menu, for example:

    ```
    Access selection menu:

           a: Admin CLI
    ```

```
       1: Serial: port1        (9600,8,1,none,none)
       q: Quit

Select access or quit [admin] :
```

Type **a** or **admin** to access the AnywhereUSB Plus command line.

You will now be connected to the Admin CLI:

```
Connecting now, 'exit' to disconnect from Admin CLI ...

>
```

See Command line interface for detailed instructions on using the command line interface.

Exit the command line interface

Command line

1. At the command prompt, type **exit**.

```
> exit
```

2. Depending on the device configuration, you may be presented with another menu, for example:

```
Access selection menu:

       a: Admin CLI
       1: Serial: port1        (9600,8,1,none,none)
       q: Quit

Select access or quit [admin] :
```

Type **q** or **quit** to exit.

Use the local REST API to configure the AnywhereUSB Plus device

Your AnywhereUSB Plus device includes a REST API that can be used to return information about the device's configuration and to make modifications to the configuration. You can view the REST API specification from your web browser by opening the URL:

https://*ip-address*/cgi-bin/config.cgi

For example:

https://192.168.210.1/cgi-bin/config.cgi

Use the GET method to return device configuration information

To return device configuration, issue the **GET** method. For example, using **curl**:

```
$ curl -k -u admin https://ip-address/cgi-bin/config.cgi/value/path -X GET
```

where:

- *ip-address* is the IP address of the AnywhereUSB Plus device.
- *path* is the path location in the configuration for the information being returned.

To determine allowed values for *path* from the Admin CLI:

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. At the config prompt, type **?** (question mark):

   ```
   (config)> ?
   auth                      Authentication
   cloud                     Central management
   firewall                  Firewall
   monitoring                Monitoring
   network                   Network
   serial                    Serial
   service                   Services
   system                    System
   vpn                       VPN

   (config)>
   ```

   The allowed values for *path* are listed in the first (left) column.

4. To determine further allowed path location values by using the **?** (question mark) with the path name:

   ```
   (config> service ?

   Services

    Additional Configuration
    ---------------------------------------------------------------------
   ----------
    dns                       DNS
    iperf                     IPerf
    location                  Location
    mdns                      Service Discovery (mDNS)

    multicast                 Multicast
    ntp                       NTP
    ping                      Ping responder
    snmp                      SNMP
    ssh                       SSH
    telnet                    Telnet
    web_admin                 Web administration
   ```

```
(config)> service
```

For example, to use **curl** to return the ssh configuration:

```
$ curl -k -u admin https://192.168.210.1/cgi-bin/config.cgi/value/service/ssh -X
GET
Enter host password for user 'admin':
{
ok": true,
       "result": {
               "type": "object",
               "path": "service.ssh"
,              "collapsed": {
"acl.zone.0": "internal"
,
"acl.zone.1": "edge"
,
"acl.zone.2": "ipsec"
,
"acl.zone.3": "setup"
,
"enable": "true"
,
"key": ""
,
"mdns.enable": "true"
,
"mdns.name": ""
,
"mdns.type": "_ssh._tcp."
,
"port": "22"
,
"protocol.0": "tcp"
               }
       }
}
$
```

You can also use the **GET** method to return the configuration parameters associated with an item:

```
curl -k -u admin https://192.168.210.1/cgi-bin/config.cgi/keys/service/ssh -X GET
Enter host password for user 'admin':
{ "ok": true, "result": [ "acl", "enable", "key", "mdns", "port", "protocol" ] }
$
```

Use the POST method to modify device configuration parameters and add items to a list array

To modify configuration parameters, use the **POST** method with the **path** and **value** parameters.

```
$ curl -k -u admin "https://ip-address/cgi-
bin/config.cgi/value?path=path&value=new_value" -X POST
```

where:

- *path* is the path to the configuration parameter, in dot notation (for example, **ssh.service.enable**).
- *new_value* is the new value for the parameter.

For example, to disable the ssh service using **curl**:

```
$ curl -k -u admin "https://192.168.210.1/cgi-
bin/config.cgi/value?path=service.ssh.enable&value=false" -X POST
Enter host password for user 'admin':
{ "ok": true }
$
```

To add items to a list array, use the **POST** method with the **path** and **append** parameters. For example, to add the external firewall zone to the ssh service:

```
$ curl -k -u admin "https://192.168.210.1/cgi-
bin/config.cgi/value?path=service.ssh.acl.zone&append=true&value=external" -X
POST
Enter host password for user 'admin':
{ "ok": true, "result": "service.ssh.acl.zone.4" }
$
```

Use the DELETE method to remove items from a list array

To remove items from a list array, use the **DELETE** method. For example, using **curl**:

```
$ curl -k -u admin "https://192.168.210.1/cgi-bin/config.cgi/value?path=path
```

where *path* is the path to the list item, including the list number, in dot notation (for example, **service.ssh.acl.zone.4**).

For example, to remove the external firewall zone to the ssh service:

1. Use the **GET** method to determine the SSH service's list number for the external zone:

```
$ curl -k -u admin "https://192.168.210.1/cgi-
bin/config.cgi/value/service/ssh/acl/zone -X GET
{
       "ok": true,
       "result": {
               "type": "array",
               "path": "service.ssh.acl.zone"
,              "collapsed": {
"0": "internal"
,
"1": "edge"
,
"2": "ipsec"
,
"3": "setup"
,
"4": "external"
               }
       }
}
$
```

2. Use the **DELETE** method to remove the external zone (list item 4).

```
$ curl -k -u admin https://192.168.210.1/cgi-
bin/config.cgi/value?path=service.ssh.acl.zone.4 -X DELETE
Enter host password for user 'admin':
{ "ok": true }
$
```

Configure the AnywhereUSB in the web user interface

You can configure the AnywhereUSB Hub in the **Configuration and Management** web user interface.

Before you begin

- Install the **AnywhereUSB Manager**

  The web UI is available from a Hub listed in the **AnywhereUSB Manager**. You must install the **AnywhereUSB Manager** before you can use the web UI. See Step 3: Install the AnywhereUSB Manager.

- Change the default password for the default **admin** user

  After initial installation, you should change the default password on the Hub for the default **admin** user. This ensures that when you log into the web UI or the command line you are able to save any configuration changes you have made. See Step 7: Change the admin password on the Hub.

WARNING! If you do not replace the default password on the Hub for the default admin user, you are able to stage configuration changes, but **you will not be able to save the configuration changes**.

AnywhereUSB Configuration page

The **AnywhereUSB Configuration** page consists of all configuration options related to AnywhereUSB.

To access this page, click **System** > **Configuration** > **AnywhereUSB Configuration**.

Service Settings

Click **Service Settings** to expand this section.

Item	Description
Enable	Click **Enable** to enable AnywhereUSB.
Port	Specify the port number that is used to access the Hub. The default value is 18574. If you change the port number you must also change the corresponding port number on your computer.
Enable USB debug logging	Select this option to enable USB debug logging. This feature should only be used when working with Digi Technical Support to debug an issue.

Group Settings

Click **Group Settings** to expand this section. In this section you can name groups and assign USB ports to the groups.

For instructions, see Create groups and assign ports to the group.

Item	Description
Group Description	A free-form description of a group. You can type over the default description. One row displays for each group, and up to 24 groups are available. The **Unassigned** group row is used for any port that is not assigned to a group.
Port Assignments	Specify the USB ports in each group. Each port on a Hub can be assigned to only one group. Ports that are not assigned to a group can be put in the **Unassigned** group. Depending on your model, 2, 8, or 24 group rows are available.

Client Connections

Click **Client Connections** to expand this section and display information about the clients that can connect to the Hub.

For more information, see Configure and manage client IDs.

Item	Description
Select a client to configure	Select the existing client that you want to update or remove. ■ **Edit**: Click **Edit** to update the selected client. ■ **Remove**: Click **Remove** to remove the selected client.
Client ID	The client ID is a unique identifier assigned to a user account the first time a user logs in to a computer and opens the **AnywhereUSB Manager**. During this process, the **AnywhereUSB Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your user account with the Hub. See Configure and manage client IDs.
Certificate	The status of the certificate associated with the client ID. This certificate is used to validate your user account with the Hub. The **Certificate** value is **Unavailable** until certificates have been exchanged between the computer and the Hub. After this occurs, the **Certificate** value is updated to **Available**. See Configure a client ID.
Description	A free-form description of the client.
Group Access	The groups that this client is allowed to access. The USB ports in the group can be accessed by this user account. See Configure a client ID.
Add Client	Click **Add Client** to manually add a new client ID. See Manually add a client ID.
Automatically Register Unknown Clients	This feature is not currently implemented.
Group Access	This section is related to the **Automatically Register Unknown Clients** option, which is not currently implemented.

AnywhereUSB Status page

The **AnywhereUSB Status** page contains status information about the USB devices and groups connected to the AnywhereUSB.

You can access this page in two ways:

- Click **Dashboard**, and then click **Show Details** in the **AnywhereUSB Service** pane.
- Click **Status** > **Services** > **AnywhereUSB**.

USB Devices

Click **USB Devices** to expand this section and display information about the USB devices connected to the AnywhereUSB.

Item	Description
🔧 configuration icon	Click the 🔧 (configuration) icon in the upper right corner of the page to access the **AnywhereUSB Configuration** page. See AnywhereUSB Configuration page for more information.
Port	The number of the USB port to which the USB device is connected.
Group	The group to which the USB port is assigned.
USB	The USB technology of the connected device: 2.0, 2.1, 3.0, or 3.1.
Manufacturer	Name of the USB device manufacturer, if supplied by the device.
Product	Name of the USB product, if supplied by the device.
Serial number	The serial number of the USB device, if supplied by the device.

Client Connections

Click to expand this section and display information about the groups connected to the AnywhereUSB.

Item	Description
🔧 configuration icon	Click the 🔧 (configuration) icon in the upper right corner of the page to access the **AnywhereUSB Configuration** page. See Configure the AnywhereUSB in the web user interface for more information.
Group	A group to which the client has connected. See Connect to a group or USB device in the AnywhereUSB Manager.
Client ID	The unique identifier of the client that has connected to this group. For more information, see Client ID.
IP Address	The network address of the client's computer.

Rename a Hub and the groups in a Hub

A default name is assigned to a Hub and to the groups in the Hub. These names are associated with the physical Hub and groups on the Hub, and can be changed in theweb user interface.

Note A USB device does not have a name that can be changed. However, a local name can be assigned to a USB device in the **AnywhereUSB Manager**. See Assign a local name to a USB device.

The default Hub name and group name can be seen by every user that connects to the Hub. You can also give a Hub and groups a local name that can be see only by the user that assigns the name. See Assign a local name to a Hub and Assign a local name to a group.

Note Only administrators can rename the Hubs and the groups.

- Rename the Hub
- Rename a group

Rename the Hub

You can rename the AnywhereUSB Hub in the **Ethernet Network Configuration** page.

By default, the Hub name is the serial number assigned to the Hub. The serial number for the Hub is on the Hub's label. The Hub name displays in the **Name** field in the Hub Status pane in the **AnywhereUSB Manager**.

Note The name can consist of the following characters: 0-9, A-Z, a-z, dash (-), or period (.). You cannot use spaces, underscores (_), comma (,), forward slash (/), or ampersand (&).

1. Open the web UI.
2. Select **System** > **Configuration** > **Device Configuration**.
3. Expand **System**.
4. In the **Name** field, enter a descriptive name for the Hub. The name cannot have spaces or underscores.
5. Click **Apply**.

Rename a group

You can rename the AnywhereUSB Hub in the **AnywhereUSB** page in the web UI.

By default, a group is named "Group" appended by a consecutive number, such as Group 1, Group 2, and so on. The group name displays in the **Group Name** field in the Group Status pane in the **AnywhereUSB Manager**.

1. Open the web UI from your selected Hub.
2. Select **System** > **AnywhereUSB Configuration**.
3. Expand **Group Settings**.
4. Enter a new name for a group in the desired **Group Description** field.
5. Click **Apply** to save the changes.

Configure and manage client IDs

The client ID is a unique identifier assigned to a user account the first time a user logs in to a computer and opens the **AnywhereUSB Manager**. During this process, the **AnywhereUSB Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your user account with the Hub. For more information, see Client ID.

Manage the client IDs

For each Hub, you can view a list of client IDs that are allowed to connect to the Hub. You can manually add client IDs.

Note You can have up to 24 client IDs in the client list.

Assign client IDs to USB ports on the Hub

The client IDs are assigned to groups of USB ports on the Hub. When a computer connects to a group in the **AnywhereUSB Manager**, the computer has access to all of the ports in the group and the devices connected to those ports. No other computer is allowed to access any of the devices in the group. A computer can connect to more than one group at a time.

- Configure a client ID
- Manually add a client ID
- Remove a client ID

Configure a client ID

You can assign a descriptive name to a client ID in the client list, and update the groups the client ID is allowed to access. The client ID can access all of the ports in the specified groups, as defined in the Group Settings section.

Note If needed, you can also add additional client IDs to the list.

1. Open the web UI.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Expand the **Client Settings** section.
4. From the client list, select the client ID that you want to configure. Information about the selected client ID displays in the **Settings for Client** section.
5. Click **Edit**.
6. In the **Description** field, enter a descriptive name for the client ID.
7. Click the check box next to a group to which the computer is allowed access. As you select groups, the selected group numbers appear in the **Group Access** field in the **Settings for Client** section. You can also manually enter group numbers in the **Group Access** field.

   Note The **Certificate** value is **Unavailable** until certificates have been exchanged between the computer and the Hub. After this occurs, the **Certificate** value is updated to **Available**.

8. Click **Apply** to save the changes.

Manually add a client ID

You can manually add client IDs to the client list. When a computer searches for Hubs, any computer with a client ID on the client list can connect to the Hub.

Note You can have up to 250 client IDs in the client list.

After you have added a client ID, the certificate is unavailable until the first time a computer with the new client ID connects to the Hub. For more information about client IDs, see Client ID.

When the computer connects to the Hub for the first time, the credentials are exchanged between the computer and the Hub. After the initial connection, only that computer with the client ID and unique identity certificate is able to connect to the Hub. Any other computer with the same client ID will be rejected. For information about computers with the same client ID, see AnywhereUSB Manager client ID is not unique.

WARNING! Digi recommends that you use a private network to connect the computer to the Hub. This ensures that only clients IDs with known user credentials can connect to the Hub. The first time that a client ID on a computer connects to the Hub, the unique credentials for this known user are stored in your Hub. If do not use a private network, an unknown computer with the same client ID may happen to connect to the Hub before the known computer connects. In this case, the known computer will not be able to connect and authenticate.

1. Open the web UI.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Expand the **Client Settings** section.
4. Click **Add Client**. A new row labeled "New Client" is added to the client list and the **Settings for Client** section is populated for the new client.
5. Enter information about the client ID in the **Settings for Client "New Client"** section.
   a. In the **Client ID** field, enter the client ID for the computer.
   b. In the **Description** field, enter a descriptive name for the client ID.
   c. Click the check box next to a group to which the computer is allowed access. As you select groups, the selected group numbers appear in the **Group Access** field in the **Settings for Clients** section.

   Note The **Certificate** value is **Unavailable** until certificates have been exchanged between the computer and the Hub. After this occurs, the **Certificate** value is updated to **Available**.
6. Click **Apply**. The client ID is added to the client list.

Remove a client ID

You can remove a client ID from the client list when a user logged in to a computer should no longer have access to the Hub.

1. Open the web UI.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Expand the **Client Settings** section.
4. In the **Select a client to configure** section, select the client ID you want to remove from the list.
5. Click **Remove**. A confirmation dialog appears.
6. Click **OK**.

View Hub system information

You can view current status information about the Hub in the **Dashboard**. This page appears by default when you launch the web UI.

1. Open the web UI. The **Dashboard** appears by default when you launch the web UI.
2. In the **AnywhereUSB Service** pane, click **Show Details** to display additional information in the AnywhereUSB Status page.

USB Devices

Click **USB Devices** to expand this section and display information about the USB devices connected to the AnywhereUSB.

Item	Description
🔧 configuration icon	Click the 🔧 (configuration) icon in the upper right corner of the page to access the **AnywhereUSB Configuration** page. See AnywhereUSB Configuration page for more information.
Port	The number of the USB port to which the USB device is connected.
Group	The group to which the USB port is assigned.
USB	The USB technology of the connected device: 2.0, 2.1, 3.0, or 3.1.
Manufacturer	Name of the USB device manufacturer, if supplied by the device.
Product	Name of the USB product, if supplied by the device.
Serial number	The serial number of the USB device, if supplied by the device.

Client Connections

Click to expand this section and display information about the groups connected to the AnywhereUSB.

Item	Description
🔧 configuration icon	Click the 🔧 (configuration) icon in the upper right corner of the page to access the **AnywhereUSB Configuration** page. See Configure the AnywhereUSB in the web user interface for more information.
Group	A group to which the client has connected. See Connect to a group or USB device in the AnywhereUSB Manager.
Client ID	The unique identifier of the client that has connected to this group. For more information, see Client ID.
IP Address	The network address of the client's computer.

Configure device identity settings

You can configure the device description, contact, and location information for the Hub in the **Configuration** page. This feature is useful to identify a specific Hub when working with a large number of Hubs in multiple locations. See Configure system information.

View current connections to the Hub

You can view information about current connections to the Hub in the **AnywhereUSB Status** page. For more information, see AnywhereUSB Status page.

1. Open the web UI.
2. Select **Status** > **Services** > **AnywhereUSB**. The **AnywhereUSB Status** page appears.
    - **USB Devices**: Expand the **USB Devices** section to display information about the devices connected to the Hub.
    - **Client Connections**: Expand the **Client Connection** section to display information about the computers connected to the Hub.

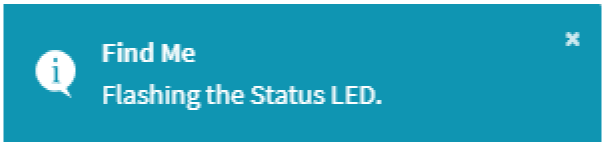
Configure the Find Me feature

You can use the Find Me feature to cause an LED on the Hub to blink, which can help you to identify a specific Hub.

- **AnywhereUSB 2 Plus**: When enabled, the power LED blinks green, then orange.
- **AnywereUSB 8 Plus** and **AnywhereUSB 24 Plus**: When enabled, the user LED blinks green, then orange.

To use this feature:

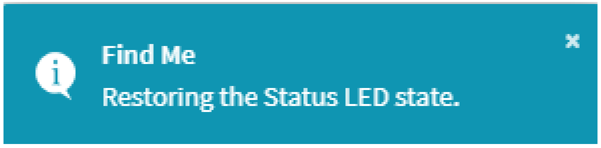
1. Open the web UI.
2. Select **System** > **Administration** > **Find Me**. A notification message appears, noting that the LED is flashing on the device. Click the X in the message to close it.



Find Me

Flashing the Status LED.

3. Select **System** > **Administration** again, and you can see that the blue circle next to **Find Me** is blinking, to alert you that the Find Me feature is active.
4. To de-activate the Find Me feature, select **System** > **Administration** > **Find Me**. A notification message appears, noting that the LED is no longer flashing on the device. Click the X in the message to close it.



Find Me

Restoring the Status LED state.

Interfaces

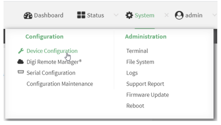
AnywhereUSB devices have several physical communications interfaces. These interfaces can be bridged in a Local Area Network (LAN) or assigned to a Wide Area Network (WAN).

This chapter contains the following topics:

Define a static IP address

You can configure a static IP address for the AnywhereUSB.

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**. The **Configuration** window is displayed.



3. Click **Network** > **Interfaces**.
4. Click the desired Ethernet section: **ETH1** or **ETH2** for AnywhereUSB Plus 24, **ETH1** for AnywhereUSB Plus 8, or **ETH** for AnywhereUSB Plus 2.

   Keep the default settings.

   - **Enable**: Selected
   - **Interface type**: Ethernet
   - **Zone**: Edge
   - **Device**: The option matches the selected Ethernet: Device: ETH1, Device: ETH2, or Device ETH.
5. Configure IPv4 settings.
   a. Click to expand **IPv4** settings.
   b. Enable IPv4 support, if it is not enabled. This is enabled by default.
   c. For **Type**, select **Static IP address**.
   d. For **Address**, type the IP address and subnet of the LAN interface. Use the format *IPv4_address/netmask*, for example, 192.168.2.1/24. For more information about the netmask, see IP address and netmask.
   e. For **Default gateway**, type the default gateway associated with this network interface.
6. (Optional) Add DNS servers to use with this static IP address.
   a. Expand the **DNS Servers** section.
   b. Click the plus sign icon next to **Add DNS server**.
   c. In the **DNS server** field, enter the IP address of the DNS server.
   d. Repeat this process if you want to add another DNS server.
7. Click **Apply** to save the configuration and apply the change.

IP address and netmask

The netmask is the length of the subnet mask in bits. For example, for a class C address with a subnet mask of 255.255.255.0, the length in bits would be 24.

NETMASK	255	255	255	255
Netmask length	8	16	24	32

Bridging

Bridging is a mechanism to create a single network consisting of multiple devices, such as Ethernet devices and wireless access points.

By default, the AnywhereUSB Plus has the following preconfigured bridges:

You can modify configuration settings for the existing bridge, and you can create new bridges.

This section contains the following topics:

Configure a bridge

Required configuration items

- A name for the bridge.

  Bridges are enabled by default.
- Devices to be included in the bridge.

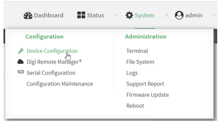
Additional configuration items

- Enable Spanning Tree Protocol (STP).

To create a bridge:

WebUI

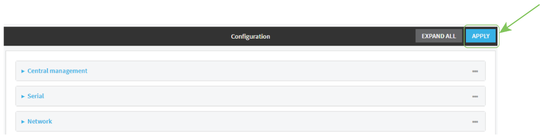
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Network** > **Bridges**.
4. For **Add Bridge**, type a name for the bridge and click ✚.
5. Bridges are enabled by default. To disable, uncheck **Enable**.
6. Add devices to the bridge:
   a. Click to expand **Devices**.
   b. For **Add device**, click ✚.
   c. Select the **Device**.
   d. Repeat to add additional devices.

   Note The MAC address of the bridge is taken from the first available device in the list.
7. (Optional) Enable Spanning Tree Protocol (STP).

   STP is used when using multiple LANs on the same device, to prevent bridge loops and other routing conflicts.
   a. Click **STP**.
   b. Click **Enable**.
   c. For **Forwarding delay**, enter the number of seconds that the device will spend in each of the listening and learning states before the bridge begins forwarding data. The default is **2** seconds.
8. Click **Apply** to save the configuration and apply the change.


Configuration
EXPAND ALL
APPLY
Central management
Serial
Network

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Create the bridge:

```
(config)> add network bridge my_bridge
(config network bridge my_bridge)>
```

4. Bridges are enabled by default.
   - To disable:

```
(config network bridge my_bridge)> enable false
(config network bridge my_bridge)>
```

   - To enable if it has been disabled:

```
(config network bridge my_bridge)> enable true
(config network bridge my_bridge)>
```

5. Add devices to the bridge:
   a. Determine available devices:

```
(config network bridge my_bridge)> .. .. interface lan device ?


Default value: /network/bridge/lan
Current value: /network/bridge/lan

(config network bridge my_bridge)>
```

   b. Add the appropriate device. For example, to add the **Digi AP** Wi-Fi access point:

```
(config network bridge my_bridge)> add device end
/network/wireless/ap/digi_ap
(config)>
```

Note The MAC address of the bridge is taken from the first available device in the list.

6. (Optional) Enable Spanning Tree Protocol (STP).

   STP is used when using multiple LANs on the same device, to prevent bridge loops and other routing conflicts.

   a. Enable STP:

```
(config network bridge my_bridge)> stp enable true
```

   b. Set the number of seconds that the device will spend in each of the listening and learning states before the bridge begins forwarding data:

      ```
      (config network bridge my_bridge)> stp forward_delay num
      (config)>
      ```

      The default is **2** seconds.

7. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Console port

AnywhereUSB Plus devices have a single serial port that provides access to the command-line interface.

Use an RS232 DB9 console cable to establish a serial connection from your AnywhereUSB Plus to your local laptop or PC. You can then use a terminal emulator program to establish the serial connection. The terminal emulator's serial connection must be configured to match the configuration of the AnywhereUSB Plus device's serial port. The default serial port configuration is:

- **Baud rate**: 115000
- **Data bits**: 8
- **Parity**: None
- **Stop bits**: 1
- **Flow control**: None

Services

This chapter contains the following topics:

Allow remote access for web administration and SSH

By default, only devices connected to the AnywhereUSB Plus's LAN have access to the device via web administration and SSH. To enable these services for access from remote devices:

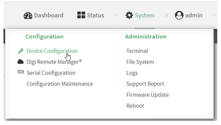
- The AnywhereUSB Plus device must have a publicly reachable IP address.
- The **External** firewall zone must be added to the web administration or SSH service. See Firewall configuration for information on zones.
- See Set the idle timeout for AnywhereUSB Plus users for information about setting the inactivity timeout for the web administration and SSH services.

To allow web administration or SSH for the External firewall zone:

Add the External firewall zone to the web administration service

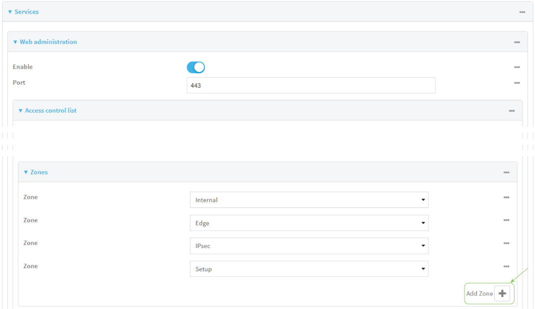
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



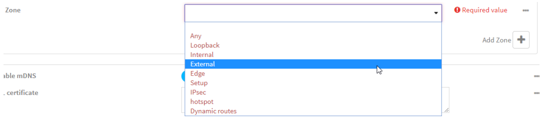
The **Configuration** window is displayed.

3. Click **Services** > **Web administration** > **Access Control List** > **Zones**.
4. For **Add Zone**, click ✚.

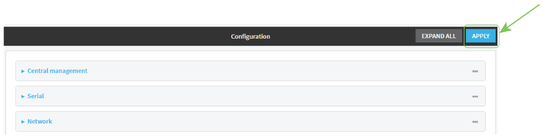


Enable
Port 443
Zone Internal
Zone Edge
Zone IPsec
Zone Setup

5. Select **External**.



6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add the external zone to the web administration service:

```
(config)> add service web_admin acl zone end external
(config)>
```

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

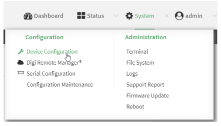
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Add the External firewall zone to the SSH service

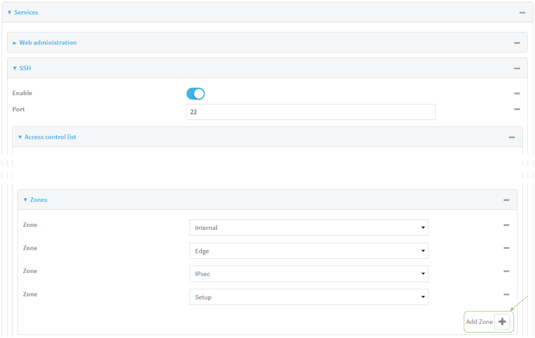
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



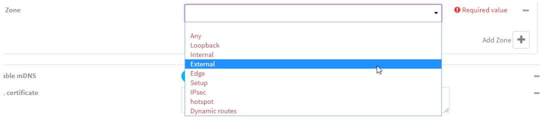
   The **Configuration** window is displayed.
3. Click **Configuration** > **Services** > **SSH** > **Access Control List** > **Zones**.

4. For **Add Zone**, click ✚.

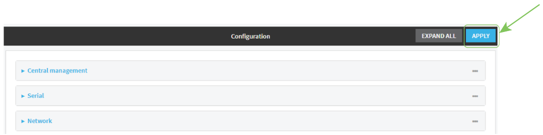


Enable ●
Port 22
Zone Internal
Zone Edge
Zone IPsec
Zone Setup

5. Select **External**.



6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add the **External** zone to the SSH service:

```
(config)> add service ssh acl zone end external
(config)>
```

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the web administration service

The web administration service allows you to monitor and configure the AnywhereUSB Plus device by using the WebUI, a browser-based interface.

By default, the web administration service is enabled and uses the standard HTTPS port, 443. The default access control for the service uses the **Internal** firewall zone, which means that only devices connected to the AnywhereUSB Plus's LAN can access the WebUI. If this configuration is sufficient for your needs, no further configuration is required. See Allow remote access for web administration and SSH for information about configuring the web administration service to allow access from remote devices.

Required configuration items

- The web administration service is enabled by default.
- Configure access control for the service.

Additional configuration items

- Port to use for web administration service communication.
- Multicast DNS (mDNS) support.
- An SSL certificate to use for communications with the service.
- Support for legacy encryption protocols.

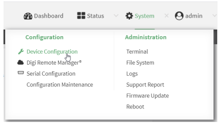
See Set the idle timeout for AnywhereUSB Plus users for information about setting the inactivity timeout for the web administration services.

Enable or disable the web administration service

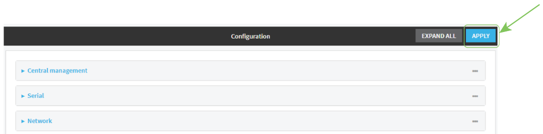
The web administration service is enabled by default. To disable the service, or enable it if it has been disabled:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Web administration**.
4. Click **Enable**.
5. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable or disable the web administration service:
   - To enable the service:

```
(config)> service web_admin enable true
(config)>
```

   - To disable the sevice:

```
(config)> service web_admin enable false
(config)>
```

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

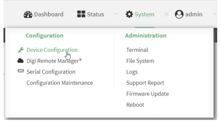
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the service

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Web administration**.
4. (Optional) For **Port**, enter the port number for the service. Normally this should not be changed.
5. Click **Access control list** to configure access control:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's web administration service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the web administration service.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:
     a. Click **IPv6 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv6 address or network that can access the device's web administration service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the web administration service.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
     a. Click **Interfaces**.
     b. For **Add Interface**, click ✚.
     c. For **Interface**, select the appropriate interface from the dropdown.
     d. Click ✚ again to allow access through additional interfaces.

- To limit access based on firewall zones:
  a. Click **Zones**.
  b. For **Add Zone**, click ✚.
  c. For **Zone**, select the appropriate firewall zone from the dropdown.
     See Firewall configuration for information about firewall zones.
  d. Click ✚ again to allow access through additional firewall zones.

6. Multicast DNS (mDNS) is enabled by default. mDNS is a protocol that resolves host names in small networks that do not have a DNS server. To disable mDNS, or enable it if it has been disabled, click **Enable mDNS**.
7. For **SSL certificate**, if you have your own signed SSL certificate, paste the certificate and private key. If **SSL certificate** is blank, the device will use an automatically-generated, self-signed certificate.
   - The SSL certificate and private key must be in PEM format.
   - The private key can use one of the following algorithms:
     - RSA
     - DSA
     - ECDSA
     - ECDH

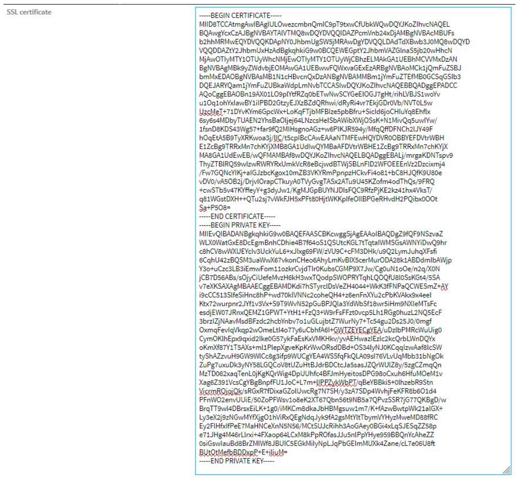
   Note Password-protected certificate keys are not supported.

   Example:

   a. Generate the SSL certificate and private key, for example:

```
# openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -
out certificate.pem
```

   b. Paste the contents of **certificate.pem** and **key.pem** into the **SSL certificate** field. The contents of the **certificate.pem** must be first. For example:



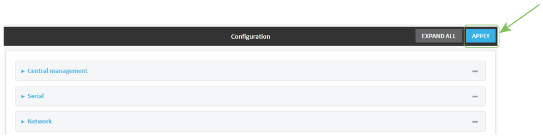
SSL certificate

```
-----BEGIN CERTIFICATE-----
MIID8TCCAtmgAwIBAgIULOwezcmbnQmIC9pT9txwCfUbkWQwDQYJKoZIhvcNAQEL
BQAwgYcxCzAJBgNVBAYTAlVTMQ8wDQYDVQQIDAZPcmVnb24xDjAMBgNVBAcMBUFs
b2hhMRMwEQYDVQQKDApNY0JhbmUgSW5jMRAwDgYDVQQLDAdTdXBwb3J0MQ8wDQYD
VQQDDAZtY2JhbmUxHzAdBgkqhkiG9w0BCQEWEGptY2JhbmVAZGlnaS5jb20wHhcN
MjAwOTIyMTY1OTUyWhcNMjEwOTIyMTY1OTUyWjCBhzELMAkGA1UEBhMCVVMxDzAN
BgNVBAgMBk9yZWdvbjEOMAwGA1UEBwwFQWxvaGExEzARBgNVBAoMCk1jQmFuZSBJ
bmMxEDAOBgNVBAsMB1N1cHBvcnQxDzANBgNVBAMMBm1jYmFuZTEfMB0GCSqGSIb3
DQEJARYQam1jYmFuZUBkaWdpLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAOBn19AX01LO9plYtfRZq0bETwNwSCYGeEIOGJ7gHt/rihLVBJS1woYv
u1Oq1ohYxlawBY1iIPBD2GtzyEJXzBZdQRhwi/dRyRi4vr7EkjGDr0Vb/NVT0L5w
UzcMeT+71DYvKYm6GpcWx+LoKqFTjbMFBIze5pbBfru+SicId6joCHIuYq8Ehflx
6sy6s4MDbyTUAEN2YhsBaOljej64LNzcsHeISbAWibXWjOSsK+N1MivQq5uwIYw/
1fsnD8KDS43Wg57+far9fQ2MIHsgnoAGz+w6PIKJR594y/MfqQffDFNCh2lJY49F
hOqEtA5B9TyXRKwoa3j/IJC/t5cpIBcCAwEAAaNTMFEwHQYDVR0OBBYEFDVtrWBH
E1ZcBg9TRRxMn7chKYjXMB8GA1UdIwQYMBaAFDVtrWBHE1ZcBg9TRRxMn7chKYjX
MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBALj/mrgaKDNTspv9
ThyZTBIRQ59wIzwRWRYRxUmkVcR8eBcjwdBTWjSBLnFID2WFOEEEnVz2Dzcixmj4
/Fw7GQNcYIKj+aIGJzbcKgox10mZB3VKYRmPpnpzHCkvFi4o81+bC8HJQfK9U80e
vDV0/vA5OB2j/DrjvlOrapCTkuyA0TVyGvgTASx2ATu9U45KZofm4odThQs/9FRQ
+cwSTb5v47KYffeyY+g3dyJw1/KgMJGpBUYNJDIsFQC9RfzPjKE2kz41hx4VksT/
q81WGstDXH++QTu2sj7vWkFJH5xPFt80HjtWKKpIfeOIlBPGeRHvdH2PQibx0OOt
Sa+P5O8=
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDgZ9fQF9NSzvaZ
WLX0WatGxE8DcEgmBnhCDhie4B7f64oS1QSUtcKGL7tTqtaIWMSGsAWNYiDwQ9hr
c8hCV8wWXUEYcIv3UckYuL6+xJIxg69FW/zVU9C+cFM3DHk/u9Q2LymJuhqXFsfi
6CqhU42zBQSM3uaWwX67vkonCHeo6AhyLmKvBIX5cerMurODA28k1ABDdmIbAWjp
Y3o+uCzc3LB3iEmwFom11ozkrCvjdTIr0KubsCGMP9X7Jw/Cg0uN1oOe/n2q/X0N
jCB7D56ABs/sOjyCiUefeMvzH6kH3wxTQodpSWOPRYTqhLQOQfU8l0SsKGt4/5SA
v7eXKSAXAgMBAAECggEBAMDKdi7hSTyrcIDsVeZH4044+WkK3fFNPaQCWESmZ+AY
i9cCC513SlfeSiHnc8hP+wd70kIVNNc2coheQH4+z6enFnXYu2cPbKVAkx9x4eeI
Ktx72wurpnr2JYf1v3Vx+S9T9WvN52pGuBPJQla3YdWbSf18wr5iHm9NXIeMTsFc
esdjEW07JRnxQEMZ1GPWT+YtH1+FzQ3+W9rFsFFzt0vcp5Lh1RGg0huzL2NQ5EcF
3brzIZjNAavMsdBFzdc2hcbYnbv7o1uGLujbtZ7WurNy7+Tc54gu2Ds25J0/0mgf
OxmqFevIqVkqp2wOmeLtl4o77y6uCbhfA6I+GWTZEYECgYEA/uDzlbPMRcWuUig0
CymOKlhEpx9qxid2Ike0G57ykFaEsKxVMKHkv/yvAEHwazIEzIc2kcQrbLWnDQYx
oKmXf87Y1T5AXs+ml1PlepXgveKpKrWwORsdDBd+OS34lyNJ0KCqqIzwAaf8lcSW
tyShAZzvuH9GW9WICc8g3ifp9WUCgYEA4WSSfqFkQLA09sI76VLvUqMbb31bNgOk
ZuPg7uxuDk3yNY58LGQCoV8tUZuHtBJdrBDCtcJa5sasJZQrWUIZ8y/5zgCZmqQn
MzTD062xaqTenL0jKgKQrWig4DpUUhfc4BFJmHyeitosDPG98oCxuh6HfuMOeM1v
Xag6Z391VcsCgYBgBnpfFU1JoC+L7m+IIPPZykWbPT/qBeYBBki5+0IhzebR9Stn
VicrmRQiojQk/sRGxR7fDixaGZolUwcRg7N7SH/y3zA7SDp4WvhjFeKFR8b6O1d4
PFnWO2envUUiE/50ZoPFWsv1o8eK2XT67Qbn56t9NB5a7QPvzSSR7jG77QKBgD/w
BrqTT9wl4DBrsxEiLK+1g0/iMKCm8dkaJbHBMgsuw1m7/K+fAzwBwtpWk21aIGX+
Ly3eX2j9zNGwMYfXjgO1hViRxQEgNdqJyk9fA2gsMtYltTbymVYHyzMweMD88fRC
Ey2FIHfxIfPeE7MaHNCeXnN5N56/MCtSUJcRihh3AoGAey0BGi4xLqSJESqZZ58p
e71JHg4M46rLlrxi+4FXaop64LCxM8kPpROfasJJu5nIPpYHye959BBQnYcAheZZ
0siGswIauBd8BrZMIWf8JBUIC5EGkMilyNpLJqPbGEImMUXk4Zane/cL7e06U8ft
BUtOtMefbBDDxpP+E+iIiuM=
-----END PRIVATE KEY-----
```

8. For **Allow legacy encryption protocols**, enable this option to allow clients to connect to the HTTPS session by using encryption protocols older than TLS 1.2, in addition to TLS 1.2 and later

protocols. This option is disabled by default, which means that only TLS 1.2 and later encryption protocols are allowed with HTTPS connections.

9. **View** is set to **Auto** by default and normally should not be changed.
10. **Legacy port redirection** is used to redirect client HTTP requests to the HTTPS service. Legacy port redirection is enabled by default, and normally these settings should not be changed. To disable legacy port redirection, click to expand **Legacy port redirection** and deselect **Enable**.
11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Configure access control:

   - To limit access to specified IPv4 addresses and networks:

```
(config)> add service web_admin acl address end value
(config)>
```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the web administratrion service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to specified IPv6 addresses and networks:

```
(config)> add service web_admin acl address6 end value
(config)>
```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the web administratrion service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

```
(config)> add service web_admin acl interface end value
(config)>
```

Where *value* is an interface defined on your device.

Display a list of available interfaces:

Use **... network interface ?** to display interface information:

```
(config)> ... network interface ?

Interfaces

Additional Configuration
-------------------------------------------
 defaultip              Default IP
 defaultlinklocal       Default Link-local IP
 eth1                   ETH1
 eth2                   ETH2
 loopback               Loopback
 modem                  Modem

(config)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service web_admin acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup
```

```
(config)>
```

Repeat this step to list additional firewall zones.

4. (Optional) If you have your own signed SSL certificate, if you have your own signed SSL certificate, set the certificate and private key by pasting their contents into the **service web_admin cert** command. Enclose the certificate and private key contents in quotes (**"**).

```
(config)> service web_admin cert "ssl-cert-and-private-key"
(config)>
```

- If **SSL certificate** is blank, the device will use an automatically-generated, self-signed certificate.
- The SSL certificate and private key must be in PEM format.
- The private key can use one of the following algorithms:
  - RSA
  - DSA
  - ECDSA
  - ECDH

  Note Password-protected certificate keys are not supported.

Example

a. Generate the SSL certificate and private key, for example:

```
# openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -
out certificate.pem
```

b. Paste the contents of **certificate.pem** and **key.pem** into the **service web_admin cert** command. Enclose the contents of **certificate.pem** and **key.pem** in quotes. For example:

```
(config)> service web_admin cert "-----BEGIN CERTIFICATE-----
MIID8TCCAtmgAwIBAgIULOwezcmbnQmIC9pT9txwCfUbkWQwDQYJKoZIhvcNAQEL
BQAwgYcxCzAJBgNVBAYTAlVTMQ8wDQYDVQQIDAZPcmVnb24xDjAMBgNVBAcMBUFs
b2hhMRMwEQYDVQQKDApNY0JhbmUgSW5jMRAwDgYDVQQLDAdTdXBwb3J0MQ8wDQYD
VQQDDAZtY2JhbmUxHzAdBgkqhkiG9w0BCQEWEGptY2JhbmVAZGlnaS5jb20wHhcN
MjAwOTIyMTY1OTUyWhcNMjEwOTIyMTY1OTUyWjCBhzELMAkGA1UEBhMCVVMxDzAN
BgNVBAgMBk9yZWdvbjEOMAwGA1UEBwwFQWxvaGExEzARBgNVBAoMCk1jQmFuZSBJ
bmMxEDAOBgNVBAsMB1N1cHBvcnQxDzANBgNVBAMMBm1jYmFuZTEfMB0GCSqGSIb3
DQEJARYQam1jYmFuZUBkaWdpLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAOBn19AX01LO9plYtfRZq0bETwNwSCYGeEIOGJ7gHt/rihLVBJS1woYv
u1Oq1ohYxIawBY1iIPBD2GtzyEJXzBZdQRhwi/dRyRi4vr7EkjGDr0Vb/NVT0L5w
UzcMeT+71DYvKYm6GpcWx+LoKqFTjbMFBIze5pbBfru+SicId6joCHIuYq8Ehflx
6sy6s4MDbyTUAEN2YhsBaOljej64LNzcsHeISbAWibXWjOSsK+N1MivQq5uwIYw/
1fsnD8KDS43Wg57+far9fQ2MIHsgnoAGz+w6PIKJR594y/MfqQffDFNCh2lJY49F
hOqEtA5B9TyXRKwoa3j/lIC/t5cpIBcCAwEAAaNTMFEwHQYDVR0OBBYEFDVtrWBH
E1ZcBg9TRRxMn7chKYjXMB8GA1UdIwQYMBaAFDVtrWBHE1ZcBg9TRRxMn7chKYjX
MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBALj/mrgaKDNTspv9
ThyZTBlRQ59wIzwRWRYRxUmkVcR8eBcjwdBTWjSBLnFlD2WFOEEEnVz2Dzcixmj4
```

```
/Fw7GQNcYIKj+aIGJzbcKgox10mZB3VKYRmPpnpzHCkvFi4o81+bC8HJQfK9U80e
vDV0/vA5OB2j/DrjvlOrapCTkuyA0TVyGvgTASx2ATu9U45KZofm4odThQs/9FRQ
+cwSTb5v47KYffeyY+g3dyJw1/KgMJGpBUYNJDIsFQC9RfzPjKE2kz41hx4VksT/
q81WGstDXH++QTu2sj7vWkFJH5xPFt80HjtWKKpIfeOIlBPGeRHvdH2PQibx0OOt
Sa+P5O8=
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDgZ9fQF9NSzvaZ
WLX0WatGxE8DcEgmBnhCDhie4B7f64oS1QSUtcKGL7tTqtaIWMSGsAWNYiDwQ9hr
c8hCV8wWXUEYcIv3UckYuL6+xJIxg69FW/zVU9C+cFM3DHk/u9Q2LymJuhqXFsfi
6CqhU42zBQSM3uaWwX67vkonCHeo6AhyLmKvBIX5cerMurODA28k1ABDdmIbAWjp
Y3o+uCzc3LB3iEmwFom11ozkrCvjdTIr0KubsCGMP9X7Jw/Cg0uN1oOe/n2q/X0N
jCB7D56ABs/sOjyCiUefeMvzH6kH3wxTQodpSWOPRYTqhLQOQfU8l0SsKGt4/5SA
v7eXKSAXAgMBAAECggEBAMDKdi7hSTyrclDsVeZH4044+WkK3fFNPaQCWESmZ+AY
i9cCC513SlfeSiHnc8hP+wd70klVNNc2coheQH4+z6enFnXYu2cPbKVAkx9x4eeI
Ktx72wurpnr2JYf1v3Vx+S9T9WvN52pGuBPJQla3YdWbSf18wr5iHm9NXIeMTsFc
esdjEW07JRnxQEMZ1GPWT+YtH1+FzQ3+W9rFsFFzt0vcp5Lh1RGg0huzL2NQ5EcF
3brzIZjNAavMsdBFzdc2hcbYnbv7o1uGLujbtZ7WurNy7+Tc54gu2Ds25J0/0mgf
OxmqFevIqVkqp2wOmeLtI4o77y6uCbhfA6I+GWTZEYECgYEA/uDzlbPMRcWuUig0
CymOKlhEpx9qxid2Ike0G57ykFaEsKxVMKHkv/yvAEHwazIEzlc2kcQrbLWnDQYx
oKmXf87Y1T5AXs+ml1PlepXgveKpKrWwORsdDBd+OS34lyNJ0KCqqIzwAaf8lcSW
tyShAZzvuH9GW9WlCc8g3ifp9WUCgYEA4WSSfqFkQLA09sI76VLvUqMbb31bNgOk
ZuPg7uxuDk3yNY58LGQCoV8tUZuHtBJdrBDCtcJa5sasJZQrWUlZ8y/5zgCZmqQn
MzTD062xaqTenL0jKgKQrWig4DpUUhfc4BFJmHyeitosDPG98oCxuh6HfuMOeM1v
Xag6Z391VcsCgYBgBnpfFU1JoC+L7m+lIPPZykWbPT/qBeYBBki5+0lhzebR9Stn
VicrmROjojQk/sRGxR7fDixaGZolUwcRg7N7SH/y3zA7SDp4WvhjFeKFR8b6O1d4
PFnWO2envUUiE/50ZoPFWsv1o8eK2XT67Qbn56t9NB5a7QPvzSSR7jG77QKBgD/w
BrqTT9wl4DBrsxEiLK+1g0/iMKCm8dkaJbHBMgsuw1m7/K+fAzwBwtpWk21alGX+
Ly3eX2j9zNGwMYfXjgO1hViRxQEgNdqJyk9fA2gsMtYltTbymVYHyzMweMD88fRC
Ey2FlHfxIfPeE7MaHNCeXnN5N56/MCtSUJcRihh3AoGAey0BGi4xLqSJESqZZ58p
e71JHg4M46rLlrxi+4FXaop64LCxM8kPpROfasJJu5nlPpYHye959BBQnYcAheZZ
0siGswIauBd8BrZMIWf8JBUIC5EGkMiIyNpLJqPbGEImMUXk4Zane/cL7e06U8ft
BUtOtMefbBDDxpP+E+iIiuM=
-----END PRIVATE KEY-----"
(config)>
```

5. (Optional) Configure Multicast DNS (mDNS):

   mDNS is a protocol that resolves host names in small networks that do not have a DNS server. mDNS is enabled by default. To disable mDNS, or enable it if it has been disabled:

   - To enable the mDNS protocol:

     ```
     (config)> service web_admin mdns enable true
     (config>
     ```

   - To disable the mDNS protocl:

     ```
     (config)> service web_admin mdns enable false
     (config)>
     ```

6. (Optional) Set the port number for this service.

The default setting of 443 normally should not be changed.

```
(config)> service web_admin port 444
(config)>
```

7. (Optional) Configure the device to allow legacy encryption protocols.

   Legacy encryption protocols allow clients to connect to the HTTPS session by using encryption protocols older than TLS 1.2, in addition to TLS 1.2 and later protocols. This option is disabled by default, which means that only TLS 1.2 and later encryption protocols are allowed with HTTPS connections.

   To enable legacy encryption protocols:

   ```
   (config)> service web_admin legacy_encryption true
   (config)>
   ```

8. (Optional) Disable legacy port redirection.

   Legacy port redirection is used to redirect client HTTP requests to the HTTPS service. Legacy port redirection is enabled by default, and normally these settings should not be changed.

   To disable legacy port redirection:

   ```
   (config)> service web_admin legacy enable false
   (config)>
   ```

9. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

10. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure SSH access

The AnywhereUSB Plus's default configuration has SSH access enabled, and allows SSH access to the device from authorized users within the **Internal** firewall zone. If this configuration is sufficient for your needs, no further configuration is required. See Allow remote access for web administration and SSH for information about configuring the SSH service to allow access from remote devices.

Required configuration items

- Enable SSH access.
- Configure access control for the SSH service.

Additional configuration items

- Port to use for communications with the SSH service.
- Multicast DNS (mDNS) support.
- A private key to use for communications with the SSH service.

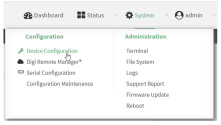
See Set the idle timeout for AnywhereUSB Plus users for information about setting the inactivity timeout for the SSH service.

Enable or disable the SSH service

The SSH service is enabled by default. To disable the service, or enable it if it has been disabled:

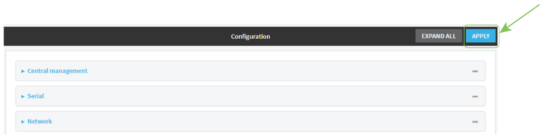
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Services** > **SSH**.
4. Click **Enable**.
5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Enable or disable the SSH service:
   - To enable the service:

     ```
     (config)> service ssh enable true
     (config)>
     ```
   - To disable the sevice:

     ```
     (config)> service ssh enable false
     (config)>
     ```
4. Save the configuration and apply the change:

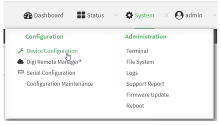
   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the service

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **SSH**.
4. (Optional) For **Port**, enter the port number for the service. Normally this should not be changed.
5. Click **Access control list** to configure access control:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.

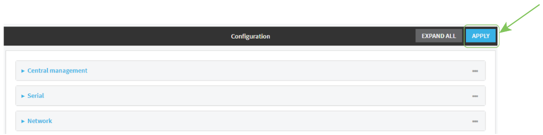
c. For **Address**, enter the IPv4 address or network that can access the device's SSH service. Allowed values are:
    - A single IP address or host name.
    - A network designation in CIDR notation, for example, 192.168.1.0/24.
    - **any**: No limit to IPv4 addresses that can access the SSH service.

d. Click ✚ again to list additional IP addresses or networks.

- To limit access to specified IPv6 addresses and networks:
    a. Click **IPv6 Addresses**.
    b. For **Add Address**, click ✚.
    c. For **Address**, enter the IPv6 address or network that can access the device's SSH service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the SSH service.
    d. Click ✚ again to list additional IP addresses or networks.
- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
    a. Click **Interfaces**.
    b. For **Add Interface**, click ✚.
    c. For **Interface**, select the appropriate interface from the dropdown.
    d. Click ✚ again to allow access through additional interfaces.
- To limit access based on firewall zones:
    a. Click **Zones**.
    b. For **Add Zone**, click ✚.
    c. For **Zone**, select the appropriate firewall zone from the dropdown.

        See Firewall configuration for information about firewall zones.
    d. Click ✚ again to allow access through additional firewall zones.

6. Multicast DNS (mDNS) is enabled by default. mDNS is a protocol that resolves host names in small networks that do not have a DNS server. To disable mDNS, or enable it if it has been disabled, click **Enable mDNS**.
7. For **Private key**, type the private key in PEM format. If **Private key** is blank, the device will use an automatically-generated key.
8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Configure access control:

   - To limit access to specified IPv4 addresses and networks:

     ```
     (config)> add service ssh acl address end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the SSH service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to specified IPv6 addresses and networks:

     ```
     (config)> add service ssh acl address6 end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the SSH service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config)> add service ssh acl interface end value
     (config)>
     ```

     Where *value* is an interface defined on your device.

     Display a list of available interfaces:

     Use **... network interface ?** to display interface information:

     ```
     (config)> ... network interface ?

     Interfaces

     Additional Configuration
     -------------------------------------------
      defaulttip             Default IP
      defaultlinklocal       Default Link-local IP
     ```

```
 eth1                   ETH1
 eth2                   ETH2
 loopback               Loopback
 modem                  Modem

(config)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service ssh acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

4. (Optional) Set the private key in PEM format. If not set, the device will use an automatically-generated key.

```
(config)> service ssh key key.pem
(config)>
```

5. (Optional) Configure Multicast DNS (mDNS)

mDNS is a protocol that resolves host names in small networks that do not have a DNS server. mDNS is enabled by default. To disable mDNS, or enable it if it has been disabled:

- To enable the mDNS protocol:

  ```
  (config)> service ssh mdns enable true
  (config>
  ```

- To disable the mDNS protocl:

  ```
  (config)> service ssh mdns enable false
  (config)>
  ```

6. (Optional) Set the port number for this service.

   The default setting of 22 normally should not be changed.

   ```
   (config)> service ssh port 24
   (config)>
   ```

7. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Use SSH with key authentication

Rather than using passwords, you can use SSH keys to authenticate users connecting via SSH, SFTP, or SCP. SSH keys provide security and scalability:

- **Security**: Using SSH keys for authentication is more secure than using passwords. Unlike a password that can be guessed by an unauthorized user, SSH key pairs provide more sophisticated security. A public key configured on the AnywhereUSB device is paired with a private key on the user's PC. The private key, once generated, remains on the user’s PC.
- **Scalability**: SSH keys can be used on more than one AnywhereUSB device.

Generating SSH key pairs

On a Microsoft Windows PC, you can generate SSH key pairs using a terminal emulator application, such as **PuTTY** or **Tera Term**.

On a Linux host, an SSH key pair is usually created automatically in the user’s **.ssh** directory. The private and public keys are named **id_rsa** and **id_rsa.pub**. If you need to generate an SSH key pair, you can use the **ssh-keygen** application.

For example, the following entry generates an RSA key pair in the user's **.ssh** directory:

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa
```

The private key file is named **id_rsa** and the public key file is named **id_rsa.pub**. (The **.pub** extension is automatically appended to the name specified for the private key output file.)

Required configuration items

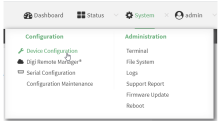
- Name for the user
- SSH public key for the user

Additional configuration items

- If you want to access the AnywhereUSB device using SSH over a WAN interface, configure the access control list for the SSH service to allow SSH access for the **External** firewall zone.

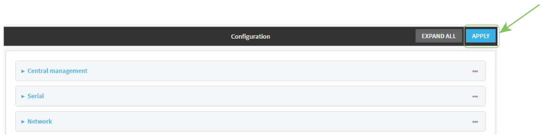
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Authentication** > **Users**.
4. Select an existing user or create a new user. See User authentication for information about creating a new user.

5. Click **SSH keys**.
6. In **Add SSH key**, enter a name for the SSH key and click ✚.
7. Enter the public SSH key by pasting or typing a public encryption key that this user can use for passwordless SSH login.
8. Click **Apply** to save the configuration and apply the change.



Command line

You can add configure passwordless SSH login for an existing user or include the support when creating a new user. See User authentication for information about creating a new user. These instructions assume an existing user named **temp_user**.

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add an SSH key for the user by using the ssh_key command and pasting or typing a public encryption key:

   ```
   (config)> add auth user maria ssh_key key_name key
   (config)>
   ```

   where:

   - *key_name* is a name for the key.
   - *key* is a public SSH key, which you can enter by pasting or typing a public encryption key that this user can use for passwordless SSH login
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure DNS

The AnywhereUSB Plus device includes a caching DNS server which forwards queries to the DNS servers that are associated with the network interfaces, and caches the results. This server is used within the device, and cannot be disabled. Use the access control list to restrict external access to this server.

Required configuration items

- Configure access control for the DNS service.

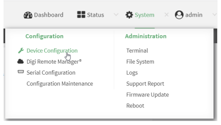
Additional configuration items

- Whether the device should cache negative responses.
- Whether the device should always perform DNS queries to all available DNS servers.
- Whether to prevent upstream DNS servers from returning private IP addresses.
- Additional DNS servers, in addition to the ones associated with the device's network interfaces.
- Specific host names and their IP addresses.

To configure the DNS server:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

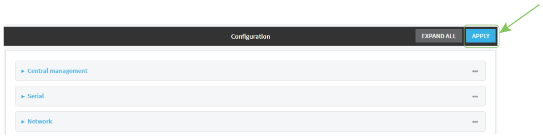
   The **Configuration** window is displayed.
3. Click **Services** > **DNS**.
4. Click **Access control list** to configure access control:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's DNS service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the DNS service.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:
     a. Click **IPv6 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv6 address or network that can access the device's DNS service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the DNS service.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
     a. Click **Interfaces**.
     b. For **Add Interface**, click ✚.
     c. For **Interface**, select the appropriate interface from the dropdown.
     d. Click ✚ again to allow access through additional interfaces.
   - To limit access based on firewall zones:
     a. Click **Zones**.
     b. For **Add Zone**, click ✚.

        c. For **Zone**, select the appropriate firewall zone from the dropdown.

           See Firewall configuration for information about firewall zones.

        d. Click ✚ again to allow access through additional firewall zones.

5. (Optional) **Cache negative responses** is enabled by default. Disabling this option may improve performance on networks with transient DNS results, when one or more DNS servers may have positive results. To disable, click **Cache negative responses**.
6. (Optional) **Query all servers** is enabled by default. This option is useful when only some DNS servers will be able to resolve hostnames. To disable, click **Query all servers**.
7. (Optional) **Rebind protection**, if enabled, prevents upstream DNS servers from returning private IP addresses. To enable, click **Rebind protection**.
8. (Optional) **Allow localhost rebinding** is enabled by default if **Rebind protection** is enabled. This is useful for Real-time Black List (RBL) servers.
9. (Optional) To add additional DNS servers:
    a. Click **DNS servers**.
    b. For **Add Server**, click ✚.
    c. (Optional) Enter a label for the DNS server.
    d. For **DNS server**, enter the IP address of the DNS server.
    e. **Domain** restricts the device's use of this DNS server based on the domain. If no domain are listed, then all queries may be sent to this server.
10. (Optional) To add host names and their IP addresses that the device's DNS server will resolve:
    a. Click **Additional DNS hostnames**.
    b. For **Add Host**, click ✚.
    c. Type the **IP address** of the host.
    d. For **Name**, type the hostname.
11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Configure access control:

   - To limit access to specified IPv4 addresses and networks:

     ```
     (config)> add service dns acl address end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the DNS service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to specified IPv6 addresses and networks:

     ```
     (config)> add service dns acl address6 end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the DNS service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config)> add service dns acl interface end value
     (config)>
     ```

     Where *value* is an interface defined on your device.

     > Display a list of available interfaces:
     >
     > Use **... network interface ?** to display interface information:

     ```
     (config)> ... network interface ?

     Interfaces

     Additional Configuration
     -------------------------------------------
      defaultip              Default IP
      defaultlinklocal       Default Link-local IP
      eth1                   ETH1
      eth2                   ETH2
      loopback               Loopback
      modem                  Modem

     (config)>
     ```

     Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

  ```
  (config)> add service dns acl zone end value
  ```

  Where *value* is a firewall zone defined on your device, or the **any** keyword.

  Display a list of available firewall zones:

  Type **... firewall zone ?** at the config prompt:

  ```
  (config)> ... firewall zone ?

  Zones: A list of groups of network interfaces that can be
  referred to by packet
  filtering rules and access control lists.

   Additional Configuration
   ---------------------------------------------------------
  ----------------------
    any
    dynamic_routes
    edge
    external
    internal
    ipsec
    loopback
    setup

  (config)>
  ```

  Repeat this step to list additional firewall zones.

4. (Optional) Cache negative responses

   By default, the device's DNS server caches negative responses. Disabling this option may improve performance on networks with transient DNS results, when one or more DNS servers may have positive results. To disable:

   ```
   (config)> service dns cache_negative_responses false
   (config>
   ```

5. (Optional) Query all servers

   By default, the device's DNS server queries all available DNS servers. Disabling this option may improve performance on networks with transient DNS results, when one or more DNS servers may have positive results. To disable:

   ```
   (config)> service dns query_all_servers false
   (config>
   ```

6. (Optional) Rebind protection

   By default, rebind protection is disabled. If enabled, this prevents upstream DNS servers from returning private IP addresses. To enable:

```
(config)> service dns stop_dns_rebind false
(config)>
```

7. (Optional) Allow localhost rebinding

   By default, localhost rebinding is enabled by default if rebind protection is enabled. This is useful for Real-time Black List (RBL) servers. To disable:

   ```
   (config)> service dns rebind_localhost_ok false
   (config)>
   ```

8. (Optional) Add additional DNS servers

   a. Add a DNS server:

      ```
      (config)> add service dns server end
      (config service dns server 0)>
      ```

   b. Set the IP address of the DNS server:

      ```
      (config service dns server 0)> address ip-addr
      (config service dns server 0)>
      ```

   c. To restrict the device's use of this DNS server based on the domain, use the **domain** command. If no domain are listed, then all queries may be sent to this server.

      ```
      (config service dns server 0)> domain domain
      (config service dns server 0)>
      ```

   d. (Optional) Set a label for this DNS server:

      ```
      (config service dns server 0)> label label
      (config service dns server 0)>
      ```

9. (Optional) Add host names and their IP addresses that the device's DNS server will resolve

   a. Add a host:

      ```
      (config)> add service dns host end
      (config service dns host 0)>
      ```

   b. Set the IP address of the host:

      ```
      (config service dns host 0)> address ip-addr
      (config service dns host 0)>
      ```

   c. Set the host name:

      ```
      (config service dns host 0)> name host-name
      (config service dns host 0)>
      ```

10. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

11. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is a protocol for remotely managing and monitoring network devices. Network administrators can use the SNMP architecture to manage nodes, including servers, workstations, routers, switches, hubs, and other equipment on an IP network, manage network performance, find and solve network problems, and plan for network growth.

The AnywhereUSB Plus device supports SNMPv3, read-only mode. SNMPv1 and v2 are not supported.

SNMP Security

By default, the AnywhereUSB Plus device automatically blocks SNMP packets from being received over WAN and LAN interfaces. As a result, if you want a AnywhereUSB Plus device to receive SNMP packets, you must configure the SNMP access control list to allow the device to receive the packets. See Configure Simple Network Management Protocol (SNMP).

Configure Simple Network Management Protocol (SNMP)

Required configuration items

- Enable SNMP.
- Firewall configuration using access control to allow remote connections to the SNMP agent.
- The user name and password used to connect to the SNMP agent.

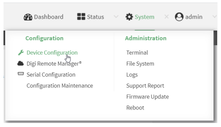
Additional configuration items

- The port used by the SNMP agent.
- Authentication type (either MD5 or SHA).
- Privacy protocol (either DES or AES).
- Privacy passphrase, if different that the SNMP user password.
- Enable Multicast DNS (mDNS) support.

To configure the SNMP agent on your AnywhereUSB Plus device:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

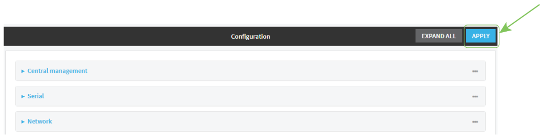


   The **Configuration** window is displayed.
3. Click **Services** > **SNMP**.
4. Click **Enable**.
5. Click **Access control list** to configure access control:

- To limit access to specified IPv4 addresses and networks:
    a. Click **IPv4 Addresses**.
    b. For **Add Address**, click ✚.
    c. For **Address**, enter the IPv4 address or network that can access the device's SNMP agent. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the SNMP agent.
    d. Click ✚ again to list additional IP addresses or networks.
- To limit access to specified IPv6 addresses and networks:
    a. Click **IPv6 Addresses**.
    b. For **Add Address**, click ✚.
    c. For **Address**, enter the IPv6 address or network that can access the device's SNMP agent. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the SNMP agent.
    d. Click ✚ again to list additional IP addresses or networks.
- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
    a. Click **Interfaces**.
    b. For **Add Interface**, click ✚.
    c. For **Interface**, select the appropriate interface from the dropdown.
    d. Click ✚ again to allow access through additional interfaces.
- To limit access based on firewall zones:
    a. Click **Zones**.
    b. For **Add Zone**, click ✚.
    c. For **Zone**, select the appropriate firewall zone from the dropdown.

        See Firewall configuration for information about firewall zones.
    d. Click ✚ again to allow access through additional firewall zones.

6. Type the **Username** used to connect to the SNMP agent.
7. Type the **Password** used to connect to the SNMP agent.
8. (Optional) For **Port**, type the port number. The default is **161**.
9. (Optional) Multicast DNS (mDNS) is disabled by default. mDNS is a protocol that resolves host names in small networks that do not have a DNS server. To enable mDNS, click **Enable mDNS**.
10. (Optional) Select the **Authentication type**, either **MD5** or **SHA**. The default is **MD5**.
11. (Optional) Type the **Privacy passphrase**. If not set, the password, entered above, is used.
12. (Optional) Select the **Privacy protocol**, either **DES** or **AES**. The default is **DES**.
13. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Enable the SNMP agent:

   ```
   (config)> service snmp enable true
   (config)>
   ```
4. Configure access control:
   - To limit access to specified IPv4 addresses and networks:

     ```
     (config)> add service snmp acl address end value
     (config)>
     ```

     Where *value* can be:
     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the SNMP service.

     Repeat this step to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:

     ```
     (config)> add service snmp acl address6 end value
     (config)>
     ```

     Where *value* can be:
     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the SNMP service.

     Repeat this step to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config)> add service snmp acl interface end value
     (config)>
     ```

     Where *value* is an interface defined on your device.

Display a list of available interfaces:

Use **... network interface ?** to display interface information:

```
(config)> ... network interface ?

Interfaces

Additional Configuration
-------------------------------------------
 defaultip               Default IP
 defaultlinklocal        Default Link-local IP
 eth1                    ETH1
 eth2                    ETH2
 loopback                Loopback
 modem                   Modem

(config)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service snmp acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ----------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

5. Set the name of the user that will be used to connect to the SNMP agent.

    ```
    (config)> service snmp username name
    (config)>
    ```

6. Set the password for the user that will be used to connect to the SNMP agent:

    ```
    (config)> service snmp password pwd
    (config)>
    ```

7. (Optional) Set the port number for the SNMP agent. The default is **161**.

    ```
    (config)> service snmp port port
    (config)>
    ```

8. (Optional) Configure Multicast DNS (mDNS)

    mDNS is a protocol that resolves host names in small networks that do not have a DNS server. For the SNMP agent, mDNS is disabled by default. To enable:

    ```
    (config)> service snmp mdns enable true
    (config>
    ```

9. (Optional) Set the authentication type. Allowed values are **MD5** or **SHA**. The default is **MD5**.

    ```
    (config)> service snmp auth_type SHA
    (config)>
    ```

10. (Optional) Set the privacy passphrase. If not set, the password, entered above, is used.

    ```
    (config)> service snmp privacy pwd
    (config)>
    ```

11. (Optional) Set the privacy protocol, either **DES** or **AES**. The default is **DES**.

    ```
    (config)> service snmp privacy_protocol AES
    (config)>
    ```

12. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

13. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Download MIBs

This procedure is available from the WebUI only.

Required configuration items

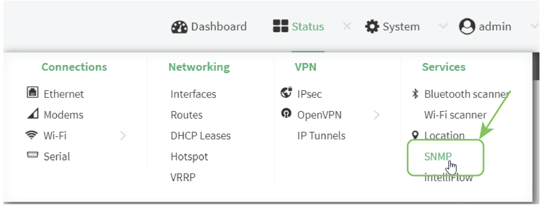
- Enable SNMP.

To download a .zip archive of the SNMP MIBs supported by this device:

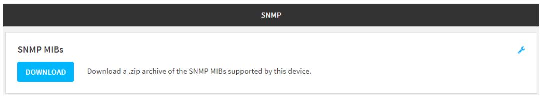
☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. Enable SNMP.

   See Configure Simple Network Management Protocol (SNMP) for information about enabling and configuring SNMP support on the AnywhereUSB Plus device.
3. On the main menu, click **Status**. Under **Services**, click **SNMP**.



   The **SNMP** page is displayed.



Download a .zip archive of the SNMP MIBs supported by this device.

4. Click **Download**.

Location information

Your AnywhereUSB Plus device can be configured to use the following location sources:

- User-defined static location.
- Location messages forwarded to the device from other location-enabled devices.

You can also configure your AnywhereUSB Plus device to forward location messages, either from the AnywhereUSB Plus device or from external sources, to a remote host. Additionally, the device can be configured to use a geofence, to allow you to determine actions that will be taken based on the physical location of the device.

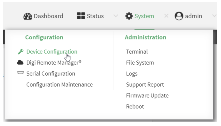
This section contains the following topics:

Configure the location service

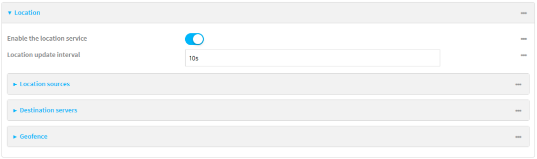
The location service is enabled by default. You can disable it, or you can enable it if it has been disabled.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Location**.

   Enable the location service ●

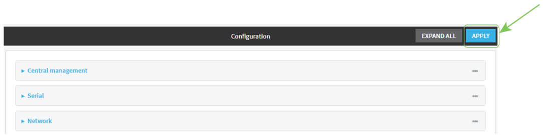
   Location update interval 10s

4. The location service is enabled by default. To disable, or to enable if it has been disabled, click **Enable**.
5. For **Location update interval**, type the amount of time to wait between polling location sources for new location data. The default is ten seconds.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **Location update interval** to ten minutes, enter **10m** or **600s**.
6. For information about configuring **Location sources**, see the following:
   a. To set a static location for the device, see Configure the device to use a user-defined static location.
   b. To accept location information from an external location-enabled server, see Configure the device to accept location messages from external sources.

   If multiple location sources are enabled at the same time, the device's location will be determined based on the order that the location sources are listed here.
7. For information about configuring **Destination servers**, see Forward location information to a remote host.
8. For information about configuring **Geofence**, see Configure geofencing.
9. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Enable or disable the GNSS module:

   - To enable the module:

     ```
     (config)> service location gnss true
     (config)>
     ```

   - To disable the module:

     ```
     (config)> service location gnss false
     (config)>
     ```

4. Set the amount of time that the AnywhereUSB Plus device will wait before polling location sources for updated location data:

   ```
   (config)> service location interval value
   (config)>
   ```

   where *value* is any number of hours, minutes, or seconds, and takes the format ***number***{**h**|**m**|**s**}.

   For example, to set **interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config)> service location interval 600s
   (config)>
   ```

   The default is 10 seconds.

5. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

6. Type **exit** to exit the Admin CLI.

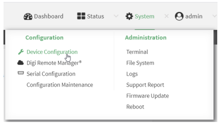
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the device to use a user-defined static location

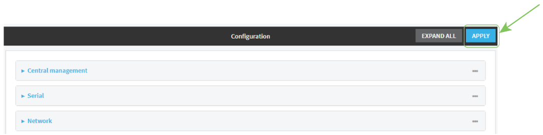
You can configured your AnywhereUSB Plus device to use a user-defined static location.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Location** > **Location sources**.
4. Click ✚ to add a location source.
5. (Optional) Type a **Label** for this location source.
6. For **Latitude**, type the latitude of the device. Allowed values are **-90** and **90**, with up to six decimal places.
7. For **Longitude**, type the longitude of the device. Allowed values are **-180** and **180**, with up to six decimal places.
8. For **Altitude**, type the altitude of the device. Allowed values are an integer followed by **m** or **km**, for example, **100m** or **1km**.
9. The location source is enabled by default. Click **Enable the location source** to disable the location source, or to enable it if it has been disabled.
10. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add a location source:

   ```
   (config)> add service location source end
   (config service location source 0)>
   ```
4. (Optional) Set a label for this location source:

   ```
   (config service location source 0)> label "label"
   (config)>
   ```

5. Set the **type** of location source to **server**:

   ```
   (config service location source 0)> type user_defined
   (config service location source 0)>
   ```

6. Set the latitude of the device:

   ```
   (config service location source 0 coordinates latitude int
   (config service location source 0)>
   ```

   where *int* is any integer between **-90** and **90**, with up to six decimal places.

7. Set the longitude of the device:

   ```
   (config service location source 0 coordinates longitude int
   (config service location source 0)>
   ```

   where *int* is any integer between **-180** and **180**, with up to six decimal places.

8. Set the altitude of the device:

   ```
   (config service location source 0 coordinates altitude alt
   (config service location source 0)>
   ```

   Where *alt* is an integer followed by **m** or **km**, for example, **100m** or **1km**.

9. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

10. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the device to accept location messages from external sources

You can configure the AnywhereUSB Plus device to accept NMEA and TAIP messages from external sources. For example, location-enabled devices connected to the AnywhereUSB Plus device can forward their location information to the device, and then the AnywhereUSB Plus device can serve as a central repository for this location information and forward it to a remote host. See Forward location information to a remote host for information about configuring the AnywhereUSB Plus device to forward location messages.

This procedure configures a UDP port on the AnywhereUSB Plus device that will be used to listen for incoming messages.

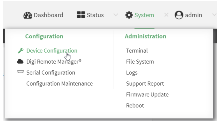
Required configuration items

- The location server must be enabled.
- UDP port that the AnywhereUSB device will listen to for incoming location messages.
- Access control list configuration to provide access to the port through the firewall.

To configure the device to accept location messages from external sources:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

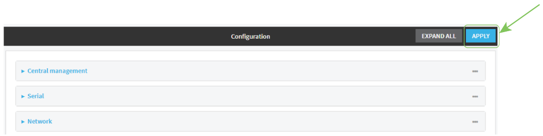
   

   The **Configuration** window is displayed.
3. Click **Services** > **Location** > **Location sources**.
4. Click ✚ to add a location source.
5. (Optional) Type a **Label** for this location source.
6. For **Type of location source**, select **Server**.
7. For **Location server port**, type the number of the UDP port that will receive incoming location messages.
8. Click **Access control list** to configure access control:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's location server UDP port. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the location server UDP port.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:
     a. Click **IPv6 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv6 address or network that can access the device's location server UDP port. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the location server UDP port.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
     a. Click **Interfaces**.
     b. For **Add Interface**, click ✚.
     c. For **Interface**, select the appropriate interface from the dropdown.
     d. Click ✚ again to allow access through additional interfaces.

- To limit access based on firewall zones:
    a. Click **Zones**.
    b. For **Add Zone**, click ✚.
    c. For **Zone**, select the appropriate firewall zone from the dropdown.

       See Firewall configuration for information about firewall zones.
    d. Click ✚ again to allow access through additional firewall zones.

9. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a location source:

```
(config)> add service location source end
(config service location source 0)>
```

4. (Optional) Set a label for this location source:

```
(config service location source 0)> label "label"
(config service location source 0)>
```

5. Set the **type** of location source to **server**:

```
(config service location source 0)> type server
(config service location source 0)>
```

6. Set the UDP port that will receive incoming location messages.

```
(config service location source 0)> server port port
(config service location source 0)>
```

7. Click **Access control list** to configure access control:
    - To limit access to specified IPv4 addresses and networks:

```
(config)> add service location source 1 acl address end value
(config)>
```

   Where *value* can be:

  - A single IP address or host name.
  - A network designation in CIDR notation, for example, 192.168.1.0/24.
  - **any**: No limit to IPv4 addresses that can access the location server UDP port.

  Repeat this step to list additional IP addresses or networks.

- To limit access to specified IPv6 addresses and networks:

```
(config)> add service location source 1 acl address6 end value
(config)>
```

  Where *value* can be:

  - A single IP address or host name.
  - A network designation in CIDR notation, for example, 2001:db8::/48.
  - **any**: No limit to IPv6 addresses that can access the location server UDP port.

  Repeat this step to list additional IP addresses or networks.

- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

```
(config)> add service location source 1 acl interface end value
(config)>
```

  Where *value* is an interface defined on your device.

  Display a list of available interfaces:

  Use **... network interface ?** to display interface information:

```
(config)> ... network interface ?

Interfaces

Additional Configuration
-------------------------------------------
 defaultip               Default IP
 defaultlinklocal        Default Link-local IP
 eth1                    ETH1
 eth2                    ETH2
 loopback                Loopback
 modem                   Modem

(config)>
```

  Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service location source 1 acl zone end value
```

  Where *value* is a firewall zone defined on your device, or the **any** keyword.

  Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

8. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

2. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Forward location information to a remote host

You can configure location clients on the AnywhereUSB Plus device that forward location messages in either NMEA or TAIP format to a remote host.

Required configuration items

- Enable the location service.
- The hostname or IP address of the remote host to which the location messages will be forwarded.
- The communication protocol, either TCP or UDP.
- The destination port on the remote host to which the messages will be forwarded.
- Message protocol type of the messages being forwarded, either NMEA or TAIP.

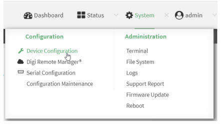
Additional configuration items

- Additional remote hosts to which the location messages will be forwarded.
- Location update interval, which determines how often the device will forward location information to the remote hosts.
- A description of the remote hosts.
- Specific types of NMEA or TAIP messages that should be forwarded.
- Text that will be prepended to the forwarded message.
- A vehicle ID that is used in the TAIP ID message and can also be prepended to the forwarded message.

Configure the AnywhereUSB device to forward location information:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

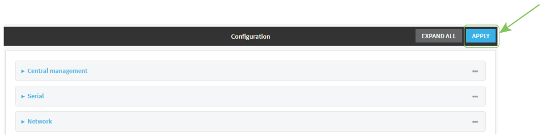
   The **Configuration** window is displayed.
3. Click **Services** > **Location** > **Destination servers**.
4. For **Add destination server**, click ✚.
5. (Optional) For **Label**, type a description of the location destination server.
6. For **Destination server**, enter the hostname or IP address of the remote host to which location messages will be sent.
7. For **Destination server port**, enter the UDP or TCP port on the remote host to which location messages will be sent.
8. For **Communication protocol**, select either **UDP** or **TCP**.
9. For **Forward interval multiplier**, select the number of **Location update intervals** to wait before forwarding location data to this server. See Configure the location service for more information about setting the **Location update interval**.
10. For **NMEA filters**, select the filters that represent the types of messages that will be forwarded. By default, all message types are forwarded.
    - To remove a filter:
      a. Click the down arrow (▾) next to the appropriate message type.
      b. Click **Delete**.
    - To add a message type:
      a. For **Add NMEA filter** or **Add TAIP filter**, click ✚.
      b. Select the filter type. Allowed values are:
         - **GGA**: Reports time, position, and fix related data.
         - **GLL**: Reports position data: position fix, time of position fix, and status.
         - **GSA**: Reports GPS DOP and active satellites.

        - **GSV**: Reports the number of SVs in view, PRN, elevation, azimuth, and SNR.
        - **RMC**: Reports position, velocity, and time.
        - **VTG**: Reports direction and speed over ground.

11. For **TAIP filters**, select the filters that represent the types of messages that will be forwarded. By default, all message types are forwarded.
    - To remove a filter:
        a. Click the down arrow (▾) next to the appropriate message type.
        b. Click **Delete**.
    - To add a message type:
        a. For **Add NMEA filter** or **Add TAIP filter**, click ✚.
        b. Select the filter type. Allowed values are:
            - **AL**: Reports altitude and vertical velocity.
            - **CP**: Compact position: reports time, latitude, and longitude.
            - **ID**: Reports the vehicle ID.
            - **LN**: Long navigation: reports the latitude, longitude, and altitude, the horizontal and vertical speed, and heading.
            - **PV**: Position/velocity: reports the latitude, longitude, and heading.
12. For **Outgoing message type**, select either NMEA or TAIP for the type of message that the device will forward to a remote host.
13. (Optional) For **Prepend text**, enter text to prepend to the forwarded message. Two variables can be included in the prepended text:
    - **%s**: Includes the AnywhereUSB device's serial number in the prepended text.
    - **%v**: Includes the vehicle ID in the prepended text.

    For example, to include both the device's serial number and vehicle ID in the prepend message, you can enter the following in the **Prepend** field:

    ```
    __|%s|__|%v|__
    ```
14. Type a four-digit alphanumeric **Vehicle ID** that will be included with to location messages. If no vehicle ID is configured, this setting defaults to 0000.
15. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a remote host to which location messages will be sent:

```
(config)> add service location forward end
(config service location forward 0)>
```

4. Set the hostname or IP address of the remote host to which location messages will be sent:

```
(config service location forward 0)> server host
(config service location forward 0)>
```

5. Set the communication protocol to either **upd** or **tcp**:

```
(config service location forward 0)> protocol protocol
(config service location forward 0)>
```

6. Set the TCP or UDP port on the remote host to which location messages will be sent:

```
(config service location forward 0)> server_port 8000
(config service location forward 0)>
```

7. Set the number of **Location update intervals** to wait before forwarding location data to this server. See Configure the location service for more information about setting the **Location update interval**.

```
(config service location forward 0)> interval_multiplier int
(config service location forward 0)>
```

8. Set the protocol type for the messages. Allowed values are **taip** or **nmea**; the default is **taip**:

```
(config service location forward 0)> type nmea
(config service location forward 0)>
```

9. (Optional) Set the text to prepend to the forwarded message. Two variables can be included in the prepended text:
   - **%s**: Includes the AnywhereUSB device's serial number in the prepended text.
   - **%v**: Includes the vehicle ID in the prepended text.

```
(config service location forward 0)> prepend __|%s|__|%v|__
(config service location forward 0)>
```

10. (Optional) Set the vehicle ID.

    Allowed value is a four digit alphanumerical string (for example, 01A3 or 1234). If no vehicle ID is configured, this setting defaults to 0000.

```
(config service location forward 0)> vehicle-id 1234
(config service location forward 0)>
```

11. (Optional) Provide a description of the remote host:

```
(config service location forward 0)> label "Remote host 1"
(config service location forward 0)>
```

12. (Optional) Specify types of messages that will be forwarded. Allowed values vary depending on the message protocol type. By default, all message types are forwarded.

    - If the message protocol type is NMEA:

      Allowed values are:

      - **gga**: Reports time, position, and fix related data.
      - **gll**: Reports position data: position fix, time of position fix, and status.
      - **gsa**: Reports GPS DOP and active satellites.
      - **gsv**: Reports the number of SVs in view, PRN, elevation, azimuth, and SNR.
      - **rmc**: Reports position, velocity, and time.
      - **vtg**: Reports direction and speed over ground.

      To remove a message type:

      a. Use the **show** command to determine the index number of the message type to be deleted:

      ```
      (config service location forward 0)> show filter_nmea
      0 gga
      1 gll
      2 gsa
      3 gsv
      4 rmc
      5 vtg
      (config service location forward 0)>
      ```

      b. Use the index number to delete the message type. For example, to delete the **gsa** (index number 2) message type:

      ```
      (config service location forward 0)> del filter_nmea 2
      (config service location forward 0)>
      ```

      To add a message type:

      a. Change to the **filter_nmea** node:

      ```
      (config service location forward 0)> filter_nmea
      (config service location forward 0 filter_nmea)>
      ```

      b. Use the **add** command to add the message type. For example, to add the **gsa** message type:

      ```
      (config service location forward 0 filter_nmea)> add gsa end
      (config service location forward 0 filter_nmea)>
      ```

    - If the message protocol type is TAIP:

      Allowed values are:

      - **al**: Reports altitude and vertical velocity.
      - **cp**: Compact position: reports time, latitude, and longitude.

- **id**: Reports the vehicle ID.
- **ln**: Long navigation: reports the latitude, longitude, and altitude, the horizontal and vertical speed, and heading.
- **pv**: Position/velocity: reports the latitude, longitude, and heading.

To remove a message type:

a. Use the **show** command to determine the index number of the message type to be deleted:

```
(config service location forward 0)> show filter_taip
0 al
1 cp
2 id
3 ln
4 pv
(config service location forward 0)>
```

b. Use the index number to delete the message type. For example, to delete the **id** (index number 2) message type:

```
(config service location forward 0)> del filter_taip 2
(config service location forward 0)>
```

To add a message type:

a. Change to the **filter_taip** node:

```
(config service location forward 0)> filter_taip
(config service location forward 0 filter_taip)>
```

b. Use the **add** command to add the message type. For example, to add the **id** message type:

```
(config service location forward 0 filter_taip)> add id end
(config service location forward 0 filter_taip)>
```

13. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

14. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure geofencing

Geofencing is a mechanism to create a virtual perimeter that allows you configure your AnywhereUSB Plus device to perform actions when entering or exiting the perimeter. For example, you can configure a device to factory default if its location service indicates that it has been moved outside of the geofence.

Multiple geofences can be defined for one device, allowing for a complex configuration in which different actions are taken depending on the physical location of the device.

Required configuration items

- Location services must be enabled.
- The geofence must be enabled.
- The boundary type of the geofence, either circular or polygonal.
  - If boundary type is circular, the latitude and longitude of the center point of the circle, and the radius.
  - If boundary type is polygonal, the latitude and longitude of the polygon's vertices (a vertex is the point at which two sides of a polygon meet). Three vertices will create a triangular polygon; four will create a square, etc. Complex polygons can be defined.
- Actions that will be taken when the device's location triggers a geofence event. You can define actions for two types of events:
  - Actions taken when the device enters the boundary of the geofence, or is inside the boundary when the device boots.
  - Actions taken when the device exits the boundary of the geofence, or is outside the boundary when the device boots.

  For each event type:

  - Determine if the action(s) associated with the event type should be performed when the device boots inside or outside of the geofence boundary.
  - The number of update intervals that should take place before the action(s) are taken.

  Multiple actions can be configured for each type of event. For each action:

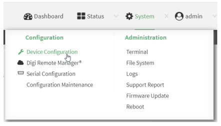
  - The type of action, either a factory erase or executing a custom script.
  - If a custom script is used:
    - The script that will be executed.
    - Whether to log output and errors from the script.
    - The maximum memory that the script will have available.
    - Whether the script should be executed within a sandbox that will prevent the script from affecting the system itself.

Additional configuration items

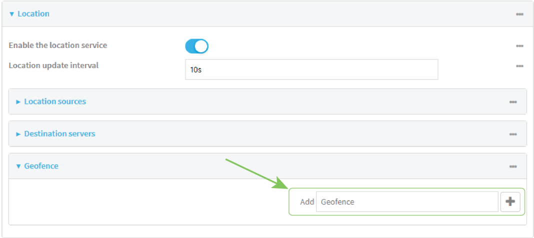
- Update interval, which determines the amount of time that the geofence should wait between polling for updated location data.

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Location** > **Geofence**.
4. For **Add Geofence**, type a name for the geofence and click ✚.

   Enable the location service

   Location update interval 10s

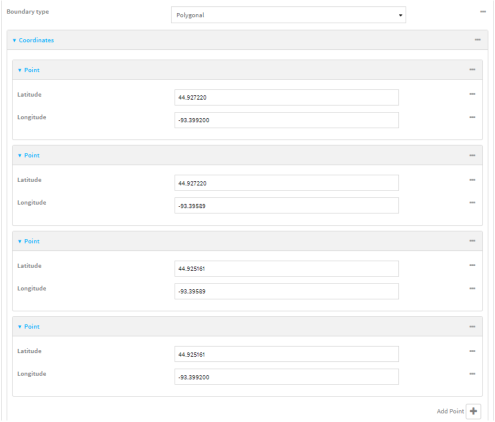
   The geofence is enabled by default. Click **Enable** to disable, or to enable if it has been disabled.
5. For **Update interval**, type the amount of time that the geofence should wait between polling for updated location data. The default is one minute.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **Update interval** to ten minutes, enter **10m** or **600s**.
6. For **Boundary type**, select the type of boundary that the geofence will have.
   - If **Circular** is selected:
     a. Click to expand **Center**.
     b. Type the **Latitude** and **Longitude** of the center point of the circle. Allowed values are:
        - For **Latitude**, any integer between **-90** and **90**, with up to six decimal places.
        - For **Longitude**, any integer between **-180** and **180**, with up to six decimal places.
     c. For **Radius**, type the radius of the circle. Allowed values are an integer followed by **m** or **km**, for example, **100m** or **1km**.
   - If **Polygonal** is selected:
     a. Click to expand **Coordinates**.
     b. Click ✚ to add a point that represents a vertex of the polygon. A vertex is the point at which two sides of a polygon meet.
     c. Type the **Latitude** and **Longitude** of one of the vertices of the polygon. Allowed values are:
        - For **Latitude**, any integer between **-90** and **90**, with up to six decimal places.
        - For **Longitude**, any integer between **-180** and **180**, with up to six decimal places.

d. Click ✚ again to add an additional point, and continue adding points to create the desired polygon.

For example, to configure a square polygon around the Digi headquarters, configure a polygon with four points:



Boundary type: Polygonal

▾ Coordinates

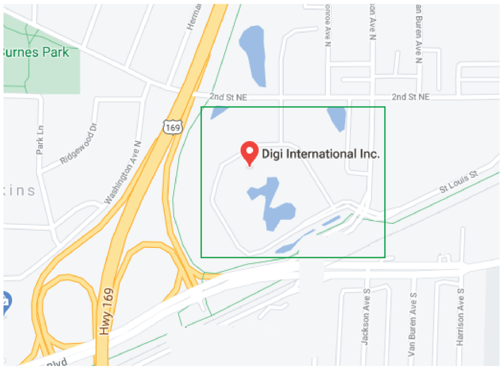
▾ Point
Latitude: 44.927220
Longitude: -93.399200

▾ Point
Latitude: 44.927220
Longitude: -93.39589

▾ Point
Latitude: 44.925161
Longitude: -93.39589

▾ Point
Latitude: 44.925161
Longitude: -93.399200

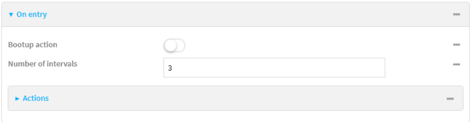
This defines a square-shaped polygon equivalent to the following:



7. Define actions to be taken when the device's location triggers a geofence event:

   - To define actions that will be taken when the device enters the geofence, or is inside the geofence when it boots:

     a. Click to expand **On entry**.



▾ On entry
Bootup action: ○
Number of intervals: 3
▸ Actions

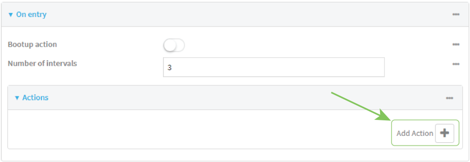
     b. (Optional) Enable **Bootup action** to configure the device to perform the **On entry** actions if the device is inside the geofence when it boots.

     c. For **Number of intervals**, type or select the number of **Update Intervals** that must take place prior to performing the **On entry** actions.

        For example, if the **Update interval** is **1m** (one minute) and the **Number of intervals** is **3**, the **On entry** actions will not be performed until the device has been inside the geofence for three minutes.

d. Click to expand **Actions**.

e. Click ✚ to create a new action.



f. For Action type, select either:

- **Factory erase** to erase the device configuration when the action is triggered.
- **Custom script** to execute a custom script when the action is triggered.

If **Custom script** is selected:

i. Click to expand **Custom script**.

ii. For **Commands**, type the script that will be executed when the action is triggered. If the script begins with **#!**, then the proceeding file path will be used to invoke the script interpreter. If not, then the default shell will be used.

iii. Enable **Log script output** to log the output of the script to the system log.

iv. Enable **Log script errors** to log errors from the script to the system log.

v. (Optional) For **Maximum memory**, type the maximum amount of system memory that will be available for the script and it spawned processes.

Allowed values are any integer followed by one of the following: **b**|**bytes**|**KB**|**k**|**MB**|**M**|**GB**|**G**|**TB**|**T**.

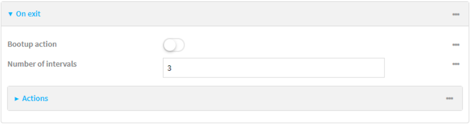
For example. the allocate one megabyte of memory to the script and its spawned processes, type **1MB** or **1M**.

vi. **Sandbox** is enabled by default. This prevents the script from adversely affecting the system. If you disable **Sandbox**, the script may render the system unusable.

vii. Repeat for any additional actions.

- To define actions that will be taken when the device exits the geofence, or is outside the geofence when it boots:

a. Click to expand **On exit**.



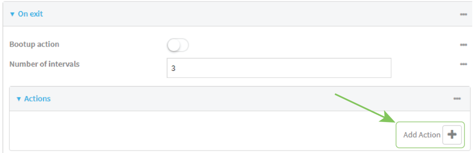
b. (Optional) Enable **Bootup action** to configure the device to perform the **On exit** actions if the device is inside the geofence when it boots.

c. For **Number of intervals**, type or select the number of **Update Intervals** that must take place prior to performing the **On exit** actions.

For example, if the **Update interval** is **1m** (one minute) and the **Number of intervals** is **3**, the **On entry** actions will not be performed until the device has been inside the geofence for three minutes.

d. Click to expand **Actions**.

e. Click ✚ to create a new action.



f. For Action type, select either:

- **Factory erase** to erase the device configuration when the action is triggered.
- **Custom script** to execute a custom script when the action is triggered.

If **Custom script** is selected:

i. Click to expand **Custom script**.

ii. For **Commands**, type the script that will be executed when the action is triggered. If the script begins with **#!**, then the proceeding file path will be used to invoke the script interpreter. If not, then the default shell will be used.

iii. Enable **Log script output** to log the output of the script to the system log.

iv. Enable **Log script errors** to log errors from the script to the system log.

v. (Optional) For **Maximum memory**, type the maximum amount of system memory that will be available for the script and it spawned processes.

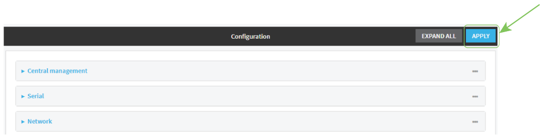
Allowed values are any integer followed by one of the following: **b**|**bytes**|**KB**|**k**|**MB**|**M**|**GB**|**G**|**TB**|**T**.

For example. the allocate one megabyte of memory to the script and its spawned processes, type **1MB** or **1M**.

vi. **Sandbox** is enabled by default. This prevents the script from adversely affecting the system. If you disable **Sandbox**, the script may render the system unusable.

vii. Repeat for any additional actions.

8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a geofence:

   ```
   (config)> add service location geofence name
   (config service location geofence name)>
   ```

   where *name* is a name for the geofence. For example:

   ```
   (config)> add service location geofence test_geofence
   (config service location geofence test_geofence)>
   ```

   The geofence is enabled by default. To disable:

   ```
   (config service location geofence test_geofence)> enable false
   (config service location geofence test_geofence)>
   ```

4. Set the amount of time that the geofence should wait between polling for updated location data:

   ```
   (config service location geofence test_geofence)> update_interval value
   (config service location geofence test_geofence)>
   ```

   where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **update_interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config service location geofence test_geofence)> update_interval 600s
   (config service location geofence test_geofence)>
   ```

   The default is **1m** (one minute).

5. Set the boundary type for the geofence:

   ```
   (config service location geofence test_geofence)> boundary value
   (config service location geofence test_geofence)>
   ```

   where *value* is either **circular** or **polygonal**.

   - If **boundary** is set to **circular** :

     a. Set the latitude and longitude of the center point of the circle:

     ```
     (config service location geofence test_geofence)> center latitude
     int
     (config service location geofence test_geofence)> center longitude
     int
     (config service location geofence test_geofence)>
     ```

     where *int* is:

     - For **latitude**, any integer between **-90** and **90**, with up to six decimal places.
     - For **longitude**, any integer between **-180** and **180**, with up to six decimal places.

b. Set the radius of the circle:

```
(config service location geofence test_geofence)> radius radius
(config service location geofence test_geofence)>
```

where *radius* is an integer followed by **m** or **km**, for example, **100m** or **1km**.

- If **boundary** is set to **polygonal**:

a. Set the coordinates of one vertex of the polygon. A vertex is the point at which two sides of a polygon meet.

i. Add a vertex:

```
(config service location geofence test_geofence)> add
coordinates end
(config service location geofence test_geofence coordinates
0)>
```

ii. Set the latitude and longitude of the vertex:

```
(config service location geofence test_geofence coordinates
0)> latitude int
(config service location geofence test_geofence coordinates
0)> longitude int
(config service location geofence test_geofence coordinates
0)>
```

where *int* is:

- For **latitude**, any integer between **-90** and **90**, with up to six decimal places.
- For **longitude**, any integer between **-180** and **180**, with up to six decimal places.

iii. Configure additional vortices:

```
(config service location geofence test_geofence coordinates
0)> ..
(config service location geofence test_geofence coordinates)>
add end
(config service location geofence test_geofence coordinates
1)> latitude int
(config service location geofence test_geofence coordinates
1)> longitude int
(config service location geofence test_geofence coordinates
1)>
```

where *int* is:

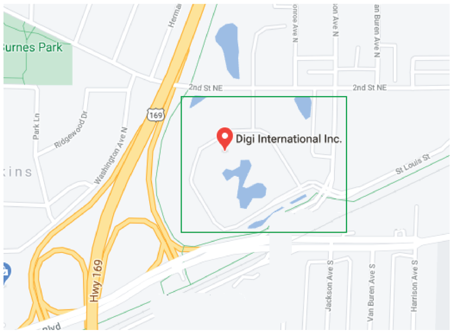
- For **latitude**, any integer between **-90** and **90**, with up to six decimal places.
- For **longitude**, any integer between **-180** and **180**, with up to six decimal places.

Repeat for each vortex of the polygon.

For example, to configure a square polygon around the Digi headquarters, configure a polygon with four points:

```
(config service location geofence test_geofence)> add
coordinates end
(config service location geofence test_geofence coordinates
0)> latitude 44.927220
(config service location geofence test_geofence coordinates
0)> longitude -93.399200
(config service location geofence test_geofence coordinates
0)> ..
(config service location geofence test_geofence coordinates)>
add end
(config service location geofence test_geofence coordinates
1)> latitude 44.927220
(config service location geofence test_geofence coordinates
1)> longitude -93.39589
(config service location geofence test_geofence coordinates
1)> ..
(config service location geofence test_geofence coordinates)>
add end
(config service location geofence test_geofence coordinates
2)> latitude 44.925161
(config service location geofence test_geofence coordinates
2)> longitude -93.39589
(config service location geofence test_geofence coordinates
2)> ..
(config service location geofence test_geofence coordinates)>
add end
(config service location geofence test_geofence coordinates
3)> latitude 44.925161
(config service location geofence test_geofence coordinates
3)> longitude -93.399200
(config service location geofence test_geofence coordinates
3)>
```

This defines a square-shaped polygon equivalent to the following:



6. Define actions to be taken when the device's location triggers a geofence event:
   - To define actions that will be taken when the device enters the geofence, or is inside the geofence when it boots:

a. (Optional) Configure the device to preform the actions if the device is inside the geofence when it boots:

```
(config)> service location geofence test_geofence on_entry bootup
true
(config)>
```

b. Set the number of update_intervals that must take place prior to performing the actions:

```
(config)> service location geofence test_geofence on_entry num_
intervals int
(config)>
```

For example, if the update interval is **1m** (one minute) and the **num_intervals** is set to **3**, the actions will not be performed until the device has been inside the geofence for three minutes.

c. Add an action:

   i. Type **...** to return to the root of the configuration:

   ```
   (config service location geofence test_geofence coordinates
   3)> ...
   (config)>
   ```

   ii. Add the action:

   ```
   (config)> add service location geofence test_geofence on_entry
   action end
   (config service location geofence test_geofence on_entry
   action 0)>
   ```

d. Set the type of action:

```
(config service location geofence test_geofence on_entry action
0)> type value
(config service location geofence test_geofence on_entry action
0)>
```

where *value* is either:

- **factory_erase**—Erases the device configuration when the action is triggered.
- **script**—Executes a custom script when the action is triggered.

factory_erase or **script**.

If **type** is set to **script**:

   i. Type or paste the script, closed in quote marks:

   ```
   (config service location geofence test_geofence on_entry
   action 0)> commands "script"
   (config service location geofence test_geofence on_entry
   action 0)>
   ```

If the script begins with **#!**, then the proceeding file path will be used to invoke the script interpreter. If not, then the default shell will be used.

ii. To log the output of the script to the system log:

```
(config service location geofence test_geofence on_entry
action 0)> syslog_stdout true
(config service location geofence test_geofence on_entry
action 0)>
```

iii. To log the errors from the script to the system log:

```
(config service location geofence test_geofence on_entry
action 0)> syslog_stderr true
(config service location geofence test_geofence on_entry
action 0)>
```

iv. (Optional) Set the maximum amount of system memory that will be available for the script and it spawned processes:

```
(config service location geofence test_geofence on_entry
action 0)> max_memory value
(config service location geofence test_geofence on_entry
action 0)>
```

where *value* is any integer followed by one of the following: **b**|**bytes**|**KB**|**k**|**MB**|**M**|**GB**|**G**|**TB**|**T**.

For example. the allocate one megabyte of memory to the script and its spawned processes:

```
(config service location geofence test_geofence on_entry
action 0)> max_memory 1MB
(config service location geofence test_geofence on_entry
action 0)>
```

v. A sandbox is enabled by default to prevent the script from adversely affecting the system. To disable the sandbox:

```
(config service location geofence test_geofence on_entry
action 0)> sandbox false
(config service location geofence test_geofence on_entry
action 0)>
```

If you disable the sandbox, the script may render the system unusable.

vi. Repeat for any additional actions.

- To define actions that will be taken when the device exits the geofence, or is outside the geofence when it boots:

  a. (Optional) Configure the device to preform the actions if the device is outside the geofence when it boots:

```
(config)> service location geofence test_geofence on_exit bootup
true
(config)>
```

b. Set the number of update_intervals that must take place prior to performing the actions:

```
(config)> service location geofence test_geofence on_exit num_
intervals int
(config)>
```

For example, if the update interval is **1m** (one minute) and the **num_intervals** is set to **3**, the actions will not be performed until the device has been outside the geofence for three minutes.

c. Add an action:

   i. Type **...** to return to the root of the configuration:

```
(config service location geofence test_geofence coordinates
3)> ...
(config)>
```

   ii. Add the action:

```
(config)> add service location geofence test_geofence on_exit
action end
(config service location geofence test_geofence on_exit action
0)>
```

d. Set the type of action:

```
(config service location geofence test_geofence on_exit action 0)>
type value
(config service location geofence test_geofence on_exit action 0)>
```

where *value* is either:

- **factory_erase**—Erases the device configuration when the action is triggered.
- **script**—Executes a custom script when the action is triggered.

factory_erase or **script**.

If **type** is set to **script**:

   i. Type or paste the script, closed in quote marks:

```
(config service location geofence test_geofence on_exit action
0)> commands "script"
(config service location geofence test_geofence on_exit action
0)>
```

If the script begins with **#!**, then the proceeding file path will be used to invoke the script interpreter. If not, then the default shell will be used.

ii. To log the output of the script to the system log:

```
(config service location geofence test_geofence on_exit action
0)> syslog_stdout true
(config service location geofence test_geofence on_exit action
0)>
```

iii. To log the errors from the script to the system log:

```
(config service location geofence test_geofence on_exit action
0)> syslog_stderr true
(config service location geofence test_geofence on_exit action
0)>
```

iv. (Optional) Set the maximum amount of system memory that will be available for the script and it spawned processes:

```
(config service location geofence test_geofence on_exit action
0)> max_memory value
(config service location geofence test_geofence on_exit action
0)>
```

where *value* is any integer followed by one of the following: **b**|**bytes**|**KB**|**k**|**MB**|**M**|**GB**|**G**|**TB**|**T**.

For example. the allocate one megabyte of memory to the script and its spawned processes:

```
(config service location geofence test_geofence on_exit action
0)> max_memory 1MB
(config service location geofence test_geofence on_exit action
0)>
```

v. A sandbox is enabled by default to prevent the script from adversely affecting the system. To disable the sandbox:

```
(config service location geofence test_geofence on_exit action
0)> sandbox false
(config service location geofence test_geofence on_exit action
0)>
```

If you disable the sandbox, the script may render the system unusable.

vi. Repeat for any additional actions.

7. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show location information

You can view status and statistics about location information from either the WebUI or the command line.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. At the **Status** page, click **Location**.

   The device's current location is displayed, along with the status of any configured geofences.

Command line

Show location information

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Use the show location command at the system prompt:

```
> show location

 Location Status
 ---------------
 State              : enabled
 Source             : 192.168.2.3
 Latitude           : 44* 55' 14.809" N (44.92078)
 Longitude          : 93* 24' 47.262" w (-93.413128)
 Altitude           : 279 meters
 Velocity           : 0 meters per second
 Direction          : None
 Quality            : Standard GNSS (2D/3D)
 UTC Date and Time  : Fri, 26 Feb 2021 8:04:23 03
 No. of Satellites  : 7

>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show geofence information

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Use the show location geofence command at the system prompt:

```
> show location geofence

 Geofence       Status  State   Transitions  Last Transition
 -------------  ------  ------  -----------  ---------------
 test_geofence  Up      Inside  0

>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

System time

By default, the AnywhereUSB Plus device synchronizes the system time by periodically connecting to the Digi NTP server, **time.devicecloud.com**. In this mode, the device queries the time server based on following events and schedule:

- At boot time.
- Once a day.

The default configuration has the system time zone set to UTC. No additional configuration is required for the system time if the default configuration is sufficient. However, you can change the default time zone and the default NTP server, as well as configuring additional NTP servers. If multiple servers are configured, a number of time samples are obtained from each of the servers and a subset of the NTP clock filter and selection algorithms are applied to select the best of these. See Configure the system time for details about changing the default configuration.

The AnywhereUSB Plus device can also be configured to use Network Time Protocol (NTP). In this configuration, the device serves as an NTP server, providing NTP services to downstream devices. See Network Time Protocol for more information about NTP server support.

Configure the system time

This procedure is optional.

The AnywhereUSB Plus device's default system time configuration uses the Digi NTP server, **time.devicecloud.com**, and has the time zone set to **UTC**. You can change the default NTP server and the default time zone, as well as configuring additional NTP servers.

Required Configuration Items

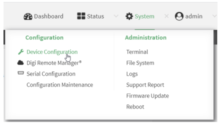
- The time zone for the AnywhereUSB Plus device.
- At least one upstream NTP server for synchronization.

Additional Configuration Options

- Additional upstream NTP servers.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



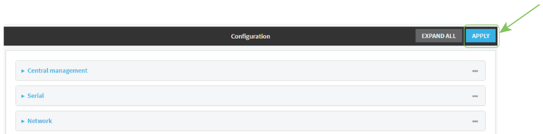
The **Configuration** window is displayed.

3. Click **System** > **Time**
4. (Optional) For **Timezone**, select either **UTC** or select the location nearest to your current location to set the timezone for your AnywhereUSB Plus device. The default is **UTC**.

5. (Optional) Add upstream NTP servers that the device will use to synchronize its time. The default setting is **time.devicecloud.com**.
    - To change the default value of the NTP server:
        a. Click **NTP servers**.
        b. For **Server**, type a new server name.
    - To add an NTP server:
        a. Click **NTP servers**.
        b. For **Add Server**, click ✚.
        c. For **Server**, enter the hostname of the upstream NTP server that the device will use to synchronize its time.
        d. Click ✚ to add additional NTP servers. If multiple servers are included, servers are tried in the order listed until one succeeds.

Note This list is synchronized with the list of servers included with NTP server configuration, and changes made to one will be reflected in the other. See Configure the device as an NTP server for more information about NTP server configuration.

6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. (Optional) Set the timezone for the location of your AnywhereUSB Plus device. The default is **UTC**.

```
(config)> system time timezone value
(config)>
```

Where *value* is the timezone using the format specified with the following command:

```
(config)> system time timezone ?

Timezone: The timezone for the location of this device. This is used to
adjust the time for log
messages. It also affects actions that occur at a specific time of day.
Format:
  Africa/Abidjan
  Africa/Accra
  Africa/Addis_Ababa
  ...

(config)>
```

4. (Optional) Add an upstream NTP server that the device will use to synchronize its time to the appropriate location in the list of NTP servers. The default setting is **time.devicecloud.com**.
   - To delete the default NTP server, **time.devicecloud.com**:

     ```
     (config)> del service ntp server 0
     ```

   - To add the NTP server to the beginning of the list, use the index value of **0** to indicate that it should be added as the first server:

     ```
     (config)> add service ntp server 0 time.server.com
     (config)>
     ```

   - To add the NTP server to the end of the list, use the index keyword **end**:

     ```
     (config)> add service ntp server end time.server.com
     (config)>
     ```

   - To add the NTP server in another location in the list, use an index value to indicate the appropriate position. For example:

     ```
     (config)> add service ntp server 1 time.server.com
     (config)>
     ```

   Note This list is synchronized with the list of servers included with NTP server configuration, and changes made to one will be reflected in the other. See Configure the device as an NTP server for more information about NTP server configuration.

5. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Network Time Protocol

Network Time Protocol (NTP) enables devices connected on local and worldwide networks to synchronize their internal software and hardware clocks to the same time source. The AnywhereUSB Plus device can be configured as an NTP server, allowing downstream hosts that are attached to the device's Local Area Networks to synchronize with the device.

When the device is configured as an NTP server, it also functions as an NTP client. The NTP client will be consistently synchronized with one or more upstream NTP servers, which means that NTP packets are transferred every few seconds. A minimum of one upstream NTP server is required. Additional NTP servers can be configured. If multiple servers are configured, a number of time samples are obtained from each of the servers and a subset of the NTP clock filter and selection algorithms are applied to select the best of these.

See Configure the device as an NTP server for information about configuring your device as an NTP server.

Configure the device as an NTP server

Required Configuration Items

- Enable the NTP service.
- At least one upstream NTP server for synchronization. The default setting is the Digi NTP server, **time.devicecloud.com**.

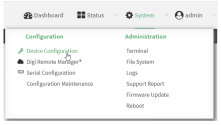
Additional Configuration Options

- Additional upstream NTP servers.
- Access control list to limit downstream access to the AnywhereUSB Plus device's NTP service.
- The time zone setting, if the default setting of UTC is not appropriate.

To configure the AnywhereUSB Plus device's NTP service:

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **NTP**.
4. Enable the AnywhereUSB Plus device's NTP service by clicking **Enable**.
5. (Optional) Configure the access control list to limit downstream access to the AnywhereUSB Plus device's NTP service.
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's NTP service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the NTP service.
     d. Click ✚ again to list additional IP addresses or networks.

- To limit access to specified IPv6 addresses and networks:
    1. Click **IPv6 Addresses**.
    2. For **Add Address**, click ✚.
    3. For **Address**, enter the IPv6 address or network that can access the device's NTP service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the NTP service.
    4. Click ✚ again to list additional IP addresses or networks.
- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
    1. Click **Interfaces**.
    2. For **Add Interface**, click ✚.
    3. For **Interface**, select the appropriate interface from the dropdown.
    4. Click ✚ again to allow access through additional interfaces.
- To limit access based on firewall zones:
    1. Click **Zones**.
    2. For **Add Zone**, click ✚.
    3. For **Zone**, select the appropriate firewall zone from the dropdown.

        See Firewall configuration for information about firewall zones.
    4. Click ✚ again to allow access through additional firewall zones.

Note By default, the access control list for the NTP service is empty, which means that all downstream hosts connected to the AnywhereUSB Plus device can use the NTP service.

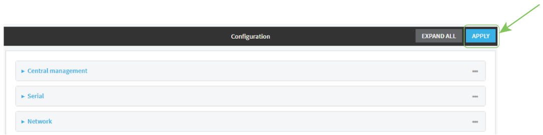
6. (Optional) Add upstream NTP servers that the device will use to synchronize its time. The default setting is **time.devicecloud.com**.
    - To change the default value of the NTP server:
        1. Click **NTP servers**.
        2. For **Server**, type a new server name.
    - To add an NTP server:
        1. Click **NTP servers**.
        2. For **Add Server**, click ✚.
        3. For **Server**, enter the hostname of the upstream NTP server that the device will use to synchronize its time.
        4. Click ✚ to add additional NTP servers. If multiple servers are included, servers are tried in the order listed until one succeeds.

Note This list is synchronized with the list of servers included with NTP client configuration, and changes made to one will be reflected in the other. See Configure the system time for more information about NTP client configuration.

7. (Optional) Configure the system time zone. The default is **UTC**.

   a. Click **System** > **Time**
   b. Select the **Timezone** for the location of your AnywhereUSB Plus device.

8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable the NTP service:

```
(config)> service NTP enable true
(config)>
```

4. (Optional) Add an upstream NTP server that the device will use to synchronize its time to the appropriate location in the list of NTP servers. The default setting is **time.devicecloud.com**.

   - To delete the default NTP server, **time.devicecloud.com**:

   ```
   (config)> del service ntp server 0
   ```

   - To add the NTP server to the beginning of the list, use the index value of **0** to indicate that it should be added as the first server:

   ```
   (config)> add service ntp server 0 time.server.com
   (config)>
   ```

   - To add the NTP server to the end of the list, use the index keyword **end**:

   ```
   (config)> add service ntp server end time.server.com
   (config)>
   ```

   - To add the NTP server in another location in the list, use an index value to indicate the appropriate position. For example:

   ```
   (config)> add service ntp server 1 time.server.com
   (config)>
   ```

   Note This list is synchronized with the list of servers included with NTP client configuration, and changes made to one will be reflected in the other. See Configure the system time for more information about NTP client configuration.

5. (Optional) Configure the access control list to limit downstream access to the AnywhereUSB Plus device's NTP service.
   - To limit access to specified IPv4 addresses and networks:

     ```
     (config)> add service ntp acl address end value
     (config)>
     ```

     Where *value* can be:
     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the NTP server agent.

     Repeat this step to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:

     ```
     (config)> add service ntp acl address6 end value
     (config)>
     ```

     Where *value* can be:
     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the NTP server agent.

     Repeat this step to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config)> add service ntp acl interface end value
     (config)>
     ```

     Where *value* is an interface defined on your device.

     Display a list of available interfaces:

     Use **... network interface ?** to display interface information:

     ```
     (config)> ... network interface ?

     Interfaces

     Additional Configuration
     -------------------------------------------
      defaultip              Default IP
      defaultlinklocal       Default Link-local IP
      eth1                   ETH1
      eth2                   ETH2
      loopback               Loopback
      modem                  Modem

     (config)>
     ```

     Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

  ```
  (config)> add service ntp acl zone end value
  ```

  Where *value* is a firewall zone defined on your device, or the **any** keyword.

  Display a list of available firewall zones:

  Type **... firewall zone ?** at the config prompt:

  ```
  (config)> ... firewall zone ?

  Zones: A list of groups of network interfaces that can be
  referred to by packet
  filtering rules and access control lists.

   Additional Configuration
   ---------------------------------------------------------
  ----------------------
    any
    dynamic_routes
    edge
    external
    internal
    ipsec
    loopback
    setup

  (config)>
  ```

  Repeat this step to list additional firewall zones.

Note By default, the access control list for the NTP service is empty, which means that all downstream hosts connected to the AnywhereUSB Plus device can use the NTP service.

6. (Optional) Set the timezone for the location of your AnywhereUSB Plus device. The default is **UTC**.

   ```
   (config)> system time timezone value
   (config)>
   ```

   Where *value* is the timezone using the format specified with the following command:

   ```
   (config)> system time timezone ?

   Timezone: The timezone for the location of this device. This is used to
   adjust the time for log
   messages. It also affects actions that occur at a specific time of day.
   Format:
     Africa/Abidjan
     Africa/Accra
     Africa/Addis_Ababa
     ...
   ```

```
(config)>
```

7. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

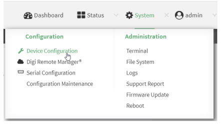
Configure a multicast route

Multicast routing allows a device to transmit data to a single multicast address, which is then distributed to a group of devices that are configured to be members of that group.

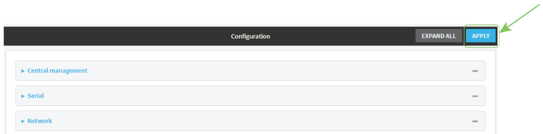
To configure a multicast route:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Multicast**.
4. For **Add Multicast route**, type a name for the route and click ✚.
5. The new route is enabled by default. To disable, uncheck **Enable**.
6. Type the **Source address** for the route. This must be a multicast IP address between 224.0.0.1 and 239.255.255.255.
7. Type the **Source port**. Ensure the port is not used by another protocol.
8. Select a **Source interface** where multicast packets will arrive.
9. Select a **Destination interface** that the AnywhereUSB Plus device will use to send mutlicast packets.
10. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Add the multicast route. For example, to add a route named **test**:

   ```
   (config)> add service multicast test
   (config service multicast test)>
   ```

4. The multicast route is enabled by default. If it has been disabled, enable the route:

```
(config service multicast test)> enable true
(config service multicast test)>
```

5. Set the source address for the route. This must be a multicast IP address between 224.0.0.1 and 239.255.255.255.

```
(config service multicast test)> dst ip-address
(config service multicast test)>
```

6. Set the source port for the route. Ensure the port is not used by another protocol.

```
(config service multicast test)> port port
(config service multicast test)>
```

7. Set the source interface for the route where multicast packets will arrive:
   a. Use the **?** to determine available interfaces:

```
(config service multicast test)> src_interface ?

Source interface: Where the multicast packets will arrive. IP routes do
not have an effect in the incoming stream.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config service multicast test)> src_interface
```

   b. Set the interface. For example:

```
(config service multicast test)> src_interface /network/interface/eth1
(config service multicast test)>
```

8. Set the destination interface that the AnywhereUSB Plus device will use to send mutlicast packets.

```
(config service multicast test)> interface interface
(config service multicast test)>
```

   a. Use the **?** to determine available interfaces:

```
(config service multicast test)> interface ?

Destination interface: Which interface to send the multicast packets.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config service multicast test)> interface
```

b. Set the interface. For example:

```
(config service multicast test)> interface /network/interface/eth1
(config service multicast test)>
```

9. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

10. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Ethernet network bonding

The AnywhereUSB Plus device supports bonding mode for the Ethernet network. This allows you to configure the device so that Ethernet ports share one IP address. When both ports are being used, they act as one Ethernet network port.

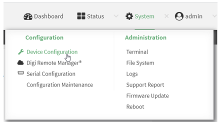
Note This applies to the AnywhereUSB 24 Plus only.

Required configuration items

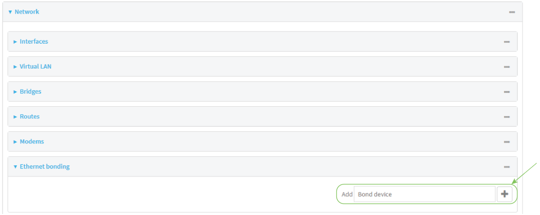
- Enable Ethernet bonding.
- The mode, either:
  - Active-backup. Provides fault tolerance.
  - Round-robin. Provides load balancing as well as fault tolerance.
- The Ethernet devices in the bonded pool.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

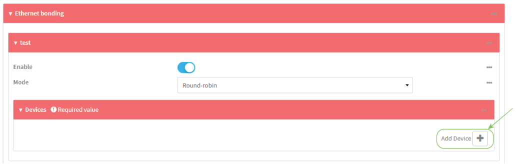
   

   The **Configuration** window is displayed.
3. Click **Network** > **Ethernet bonding**.
4. For **Add Bond device**, click ✚

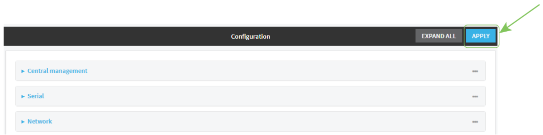
   

   The bond device is enabled by default. To disable, click to toggle off **Enable**.
5. For **Mode**, selected either:
   - **Active-backup**: Transmits data on only one of the bonded devices at a time. When the active device fails, the next available device in the list is chosen. This mode provides for fault tolerance.
   - **Round-robin**: Alternates between bonded devices to provide load balancing as well as fault tolerance.
6. Click to expand **Devices**.

7. Add Ethernet devices:
   a. For **Add device**, click ✚



   b. For **Device**, select an Ethernet device to participate in the bond pool.
   c. Repeat for each appropriate Ethernet device.
8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a network bond:

```
(config)> add network bond name
(config network bond name)>
```

   The new network bond is enabled by default. To disable:

```
(config network bond name)> enable false
(config network bond name)>
```

4. Set the mode:

```
(config network bond name)> mode value
(config network bond name)>
```

   where value is either:
   - **active-backup**: Transmits data on only one of the bonded devices at a time. When the active device fails, the next available device in the list is chosen. This mode provides for fault tolerance.
   - **round-robin**: Alternates between bonded devices to provide load balancing as well as fault tolerance.

5. Add Ethernet devices:
   a. Use the **?** to determine available devices:

```
(config network bond name)> ... network device ?

 Additional Configuration
 -------------------------------------------------------------------------
-----
 eth1
 eth2
 loopback

(config network bond name)>
```

   b. Add a device:

```
(config network bond name)> add device /network/device/eth1
(config network bond name)>
```

   c. Repeat to add additional devices.
6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

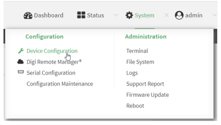
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Enable service discovery (mDNS)

Multicast DNS mDNS is a protocol that resolves host names in small networks that do not have a DNS server. You can enable the AnywhereUSB Plus device to use mDNS.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Services** > **Service Discovery (mDNS)**.
4. **Enable** the mDNS service.
5. Click **Access control list** to configure access control:

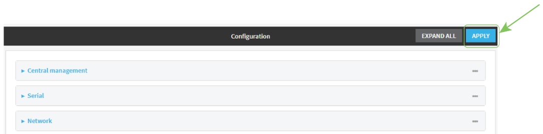
- To limit access to specified IPv4 addresses and networks:
    a. Click **IPv4 Addresses**.
    b. For **Add Address**, click ✚.
    c. For **Address**, enter the IPv4 address or network that can access the device's mDNS service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the mDNS service.

    d. Click ✚ again to list additional IP addresses or networks.
- To limit access to specified IPv6 addresses and networks:
    a. Click **IPv6 Addresses**.
    b. For **Add Address**, click ✚.
    c. For **Address**, enter the IPv6 address or network that can access the device's mDNS service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the mDNS service.

    d. Click ✚ again to list additional IP addresses or networks.
- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
    a. Click **Interfaces**.
    b. For **Add Interface**, click ✚.
    c. For **Interface**, select the appropriate interface from the dropdown.
    d. Click ✚ again to allow access through additional interfaces.
- To limit access based on firewall zones:
    a. Click **Zones**.
    b. For **Add Zone**, click ✚.
    c. For **Zone**, select the appropriate firewall zone from the dropdown.

       See Firewall configuration for information about firewall zones.
    d. Click ✚ again to allow access through additional firewall zones.

6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Enable the mDNS service:

   ```
   (config)> service mdns enable true
   (config)>
   ```

4. Configure access control:

   - To limit access to specified IPv4 addresses and networks:

     ```
     (config)> add service mdns acl address end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the mDNS service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to specified IPv6 addresses and networks:

     ```
     (config)> add service mdns acl address6 end value
     (config)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the mDNS service.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config)> add service mdns acl interface end value
     (config)>
     ```

     Where *value* is an interface defined on your device.

     Display a list of available interfaces:

     Use **... network interface ?** to display interface information:

     ```
     (config)> ... network interface ?

     Interfaces

     Additional Configuration
     -------------------------------------------
      defaultip              Default IP
     ```

```
 defaultlinklocal       Default Link-local IP
 eth1                   ETH1
 eth2                   ETH2
 loopback               Loopback
 modem                  Modem

(config)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service mdns acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Use the iPerf service

Your AnywhereUSB Plus device includes an iPerf3 server that you can use to test the performance of your network.

iPerf3 is a command-line tool that measures the maximum network throughput an interface can handle. This is useful when diagnosing network speed issues, to determine, for example, whether a cellular connection is providing expected throughput.

The AnywhereUSB Plus implementation of iPerf3 supports testing with both TCP and UDP.

Note Using iPerf clients that are at a version earlier than iPerf3 to connect to the AnywhereUSB Plus device's iPerf3 server may result in unpredictable results. As a result, Digi recommends using an iPerf client at version 3 or newer to connect to the AnywhereUSB Plus device's iPerf3 server.

Required configuration items

- Enable the iPerf server on the AnywhereUSB Plus device.
- An iPerf3 client installed on a remote host. iPerf3 software can be downloaded at https://iperf.fr/iperf-download.php.

Additional configuration Items

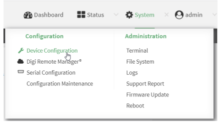
- The port that the AnywhereUSB Plus device's iPerf server will use to listen for incoming connections.
- The access control list for the iPerf server.

  When the iPerf server is enabled, the AnywhereUSB Plus device will automatically configure its firewall rules to allow incoming connections on the configured listening port. You can restrict access by configuring the access control list for the iPerf server.

To enable the iPerf3 server:

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **iPerf**.
4. Click **Enable**.
5. (Optional) For **IPerf Server Port**, type the appropriate port number for the iPerf server listening port.
6. (Optional) Click to expand **Access control list** to restrict access to the iPerf server:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's iperf service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the iperf service.

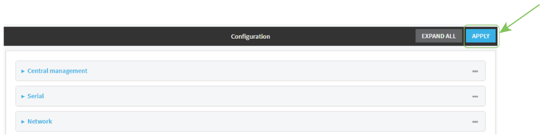
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:
     a. Click **IPv6 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv6 address or network that can access the device's iperf service. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the iperf service.

     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
     a. Click **Interfaces**.
     b. For **Add Interface**, click ✚.
     c. For **Interface**, select the appropriate interface from the dropdown.
     d. Click ✚ again to allow access through additional interfaces.

- To limit access based on firewall zones:
  a. Click **Zones**.
  b. For **Add Zone**, click ✚.
  c. For **Zone**, select the appropriate firewall zone from the dropdown.

     See Firewall configuration for information about firewall zones.
  d. Click ✚ again to allow access through additional firewall zones.

7. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable the iPerf server:

```
(config)> service iperf enable true
(config)>
```

4. (Optional) Set the port number for the iPerf server listening port. The default is 5201.

```
(config)> service iperf port port_number
(config)>
```

5. (Optional) Set the access control list to restrict access to the iPerf server:
   - To limit access to specified IPv4 addresses and networks:

```
(config)> add service iperf acl address end value
(config)>
```

     Where *value* can be:
     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the service-type.

     Repeat this step to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:

```
(config)> add service iperf acl address6 end value
(config)>
```

Where *value* can be:

- A single IP address or host name.
- A network designation in CIDR notation, for example, 2001:db8::/48.
- **any**: No limit to IPv6 addresses that can access the service-type.

Repeat this step to list additional IP addresses or networks.

- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

```
(config)> add service iperf acl interface end value
(config)>
```

Where *value* is an interface defined on your device.

Display a list of available interfaces:

Use **... network interface ?** to display interface information:

```
(config)> ... network interface ?

Interfaces

Additional Configuration
-------------------------------------------
 defaultip               Default IP
 defaultlinklocal        Default Link-local IP
 eth1                    ETH1
 eth2                    ETH2
 loopback                Loopback
 modem                   Modem

(config)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config)> add service iperf acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
```

```
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Example performance test using iPerf3

On a remote host with iPerf3 installed, enter the following command:

```
$ iperf3 -c device_ip
```

where *device_ip* is the IP address of the AnywhereUSB Plus device. For example:

```
$ iperf3 -c 192.168.2.1
Connecting to host 192.168.2.1, port 5201
[  4] local 192.168.3.100 port 54934 connected to 192.168.1.1 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  26.7 MBytes   224 Mbits/sec    8   2.68 MBytes
[  4]   1.00-2.00   sec  28.4 MBytes   238 Mbits/sec   29   1.39 MBytes
[  4]   2.00-3.00   sec  29.8 MBytes   250 Mbits/sec    0   1.46 MBytes
[  4]   3.00-4.00   sec  31.2 MBytes   262 Mbits/sec    0   1.52 MBytes
[  4]   4.00-5.00   sec  32.1 MBytes   269 Mbits/sec    0   1.56 MBytes
[  4]   5.00-6.00   sec  32.5 MBytes   273 Mbits/sec    0   1.58 MBytes
[  4]   6.00-7.00   sec  33.9 MBytes   284 Mbits/sec    0   1.60 MBytes
[  4]   7.00-8.00   sec  33.7 MBytes   282 Mbits/sec    0   1.60 MBytes
[  4]   8.00-9.00   sec  33.5 MBytes   281 Mbits/sec    0   1.60 MBytes
[  4]   9.00-10.00  sec  33.2 MBytes   279 Mbits/sec    0   1.60 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   315 MBytes   264 Mbits/sec   37             sender
[  4]   0.00-10.00  sec   313 MBytes   262 Mbits/sec                  receiver

iperf Done.
$
```

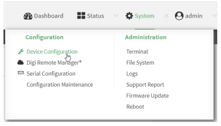
Configure the ping responder service

Your AnywhereUSB Plus device's ping responder service replies to ICMP and ICMPv6 echo requests. The service is enabled by default. You can disable the service, or you can configure the service to use an access control list to limit the service to specified IP address, interfaces, and/or zones.

To enable the iPerf3 server:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Services** > **Ping responder**.

   The ping responder service is enabled by default. Click **Enable** to disable all ping responses.
4. Click to expand **Access control list** to restrict ping responses to specified IP address, interfaces, and/or zones:
   - To limit access to specified IPv4 addresses and networks:
     a. Click **IPv4 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv4 address or network that can access the device's ping responder. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 192.168.1.0/24.
        - **any**: No limit to IPv4 addresses that can access the ping responder.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to specified IPv6 addresses and networks:
     a. Click **IPv6 Addresses**.
     b. For **Add Address**, click ✚.
     c. For **Address**, enter the IPv6 address or network that can access the device's ping responder. Allowed values are:
        - A single IP address or host name.
        - A network designation in CIDR notation, for example, 2001:db8::/48.
        - **any**: No limit to IPv6 addresses that can access the ping responder.
     d. Click ✚ again to list additional IP addresses or networks.
   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
     a. Click **Interfaces**.
     b. For **Add Interface**, click ✚.

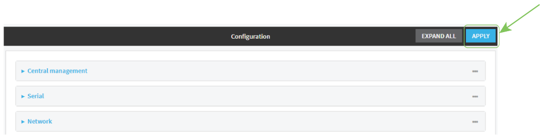
   c. For **Interface**, select the appropriate interface from the dropdown.

   d. Click ✚ again to allow access through additional interfaces.

- To limit access based on firewall zones:
  a. Click **Zones**.
  b. For **Add Zone**, click ✚.
  c. For **Zone**, select the appropriate firewall zone from the dropdown.

     See Firewall configuration for information about firewall zones.
  d. Click ✚ again to allow access through additional firewall zones.

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable the iPerf server:

```
(config)> service iperf enable true
(config)>
```

4. (Optional) Set the port number for the iPerf server listening port. The default is 5201.

```
(config)> service iperf port port_number
(config)>
```

5. (Optional) Set the access control list to restrict access to the iPerf server:
   - To limit access to specified IPv4 addresses and networks:

```
(config)> add service iperf acl address end value
(config)>
```

   Where *value* can be:
   - A single IP address or host name.
   - A network designation in CIDR notation, for example, 192.168.1.0/24.
   - **any**: No limit to IPv4 addresses that can access the service-type.

   Repeat this step to list additional IP addresses or networks.

- To limit access to specified IPv6 addresses and networks:

  ```
  (config)> add service iperf acl address6 end value
  (config)>
  ```

  Where *value* can be:

  - A single IP address or host name.
  - A network designation in CIDR notation, for example, 2001:db8::/48.
  - **any**: No limit to IPv6 addresses that can access the service-type.

  Repeat this step to list additional IP addresses or networks.

- To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

  ```
  (config)> add service iperf acl interface end value
  (config)>
  ```

  Where *value* is an interface defined on your device.

  Display a list of available interfaces:

  Use **... network interface ?** to display interface information:

  ```
  (config)> ... network interface ?

  Interfaces

  Additional Configuration
  -------------------------------------------
   defaultip               Default IP
   defaultlinklocal        Default Link-local IP
   eth1                    ETH1
   eth2                    ETH2
   loopback                Loopback
   modem                   Modem

  (config)>
  ```

  Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

  ```
  (config)> add service iperf acl zone end value
  ```

  Where *value* is a firewall zone defined on your device, or the **any** keyword.

  Display a list of available firewall zones:

  Type **... firewall zone ?** at the config prompt:

  ```
  (config)> ... firewall zone ?

  Zones: A list of groups of network interfaces that can be
  referred to by packet
  ```

```
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config)>
```

Repeat this step to list additional firewall zones.

6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Example performance test using iPerf3

On a remote host with Iperf3 installed, enter the following command:

```
$ iperf3 -c device_ip
```

where *device_ip* is the IP address of the AnywhereUSB Plus device. For example:

```
$ iperf3 -c 192.168.2.1
Connecting to host 192.168.2.1, port 5201
[  4] local 192.168.3.100 port 54934 connected to 192.168.1.1 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  26.7 MBytes   224 Mbits/sec    8   2.68 MBytes
[  4]   1.00-2.00   sec  28.4 MBytes   238 Mbits/sec   29   1.39 MBytes
[  4]   2.00-3.00   sec  29.8 MBytes   250 Mbits/sec    0   1.46 MBytes
[  4]   3.00-4.00   sec  31.2 MBytes   262 Mbits/sec    0   1.52 MBytes
[  4]   4.00-5.00   sec  32.1 MBytes   269 Mbits/sec    0   1.56 MBytes
[  4]   5.00-6.00   sec  32.5 MBytes   273 Mbits/sec    0   1.58 MBytes
[  4]   6.00-7.00   sec  33.9 MBytes   284 Mbits/sec    0   1.60 MBytes
[  4]   7.00-8.00   sec  33.7 MBytes   282 Mbits/sec    0   1.60 MBytes
[  4]   8.00-9.00   sec  33.5 MBytes   281 Mbits/sec    0   1.60 MBytes
[  4]   9.00-10.00  sec  33.2 MBytes   279 Mbits/sec    0   1.60 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   315 MBytes   264 Mbits/sec   37             sender
[  4]   0.00-10.00  sec   313 MBytes   262 Mbits/sec                  receiver

```

```
iperf Done.
$
```

User authentication

This chapter contains the following topics:

AnywhereUSB Plus user authentication

User authentication on the AnywhereUSB Plus has the following features and default configuration:

Feature	Description	Default configuration
Idle timeout	Determines how long a user session can be idle before the system automatically disconnects.	■ 10 minutes.
Allow shell	If disabled, prevents all authentication prohibits access to the shell prompt for all authentication groups. This does not prevent access to the Admin CLI. **Note** If shell access is disabled, re-enabling it will erase the device's configuration and perform a factory reset.	■ Enabled.
Methods	Determines how users are authenticated for access: **local users**, **TACACS+**, or **RADIUS**.	■ **local users**.
Groups	Associates access permissions for a group. . You can modify the released groups and create additional groups as needed for your site. A user can be assigned to more than one group.	■ **admin**: Provides the logged-in user with administrative and shell access. ■ **serial**: Provides the logged-in user with access to serial ports.
Users	Defines local users for the AnywhereUSB Plus.	■ **admin**: Belongs to both the **admin** and **serial** groups.
TACACS+	Configures support for TACACS+ (Terminal Access Controller Access-Control System Plus) servers and users.	■ Not configured.
RADIUS	Configures support for RADIUS (Remote Authentication Dial-In User Service) servers and users.	■ Not configured.
LDAP	Configures support for LDAP (Lightweight Directory Access Protocol) servers and users.	■ Not configured.

User authentication methods

Authentication methods determine how users of the AnywhereUSB Plus device are authenticated. Available authentication methods are:

- **Local users**: User are authenticated on the local device.
- **RADIUS**: Users authenticated by using a remote RADIUS server for authentication.

  See Remote Authentication Dial-In User Service (RADIUS) for information about configuring RADIUS authentication.
- **TACACS+**: Users authenticated by using a remote TACACS+ server for authentication.

  See Terminal Access Controller Access-Control System Plus (TACACS+) for information about configuring TACACS+ authentication.
- **LDAP**: Users authenticated by using a remote LDAP server for authentication.

  See LDAP for information about configuring LDAP authentication.

Add a new authentication method

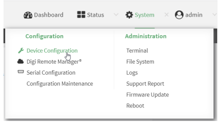
Required configuration items

- The types of authentication method to be used:

To add an authentication method:

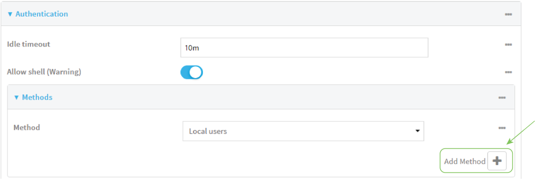
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Authentication** > **Methods**.
4. For **Add Method**, click ✚.

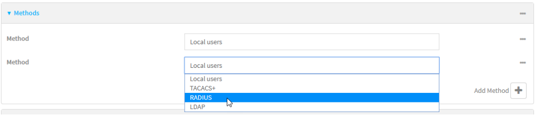


Idle timeout: 10m

Allow shell (Warning): ●

Method: Local users

5. Select the appropriate authentication type for the new method from the **Method** drop-down.

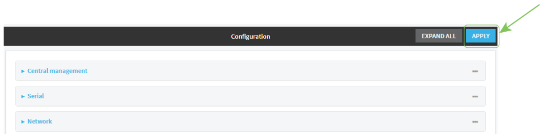


Method: Local users

Method: Local users

> **Note** Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. See Rearrange the position of authentication methods for information about how to reorder the authentication methods.

6. Repeat these steps to add additional methods.
7. Click **Apply** to save the configuration and apply the change.



Command line

Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. This procedure describes how to add methods to various places in the list.

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add the new authentication method to the appropriate location in the list:
   - To determine the current list of authentication methods:
     1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

        Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
     2. At the command line, type **config** to enter configuration mode:

        ```
        > config
        (config)>
        ```
     3. Use the **show auth method** command to display the current authentication methods configuration:

        ```
        (config)> show auth method
        0 local
        (config)>
        ```
   - To add the new authentication method to the beginning of the list, use the index value of **0** to indicate that it should be added as the first method:

     ```
     (config)> add auth method 0 auth_type
     (config)>
     ```

     where *auth_type* is one of **local**, **radius**, **tacacs+**, or **ldap**.
   - To add the new authentication method to the end of the list, use the index keyword **end**:

     ```
     (config)> add auth method end auth_type
     (config)>
     ```

     where *auth_type* is one of **local**, **radius**, **tacacs+**, or **ldap**.
   - To add the new authentication in another location in the list, use an index value to indicate the appropriate position. For example:

     ```
     (config)> add auth method 1 auth_type
     (config)>
     ```

     where *auth_type* is one of **local**, **radius**, **tacacs+**, or **ldap**.

- You can also use the **move** command to rearrange existing methods. See Rearrange the position of authentication methods for information about how to reorder the authentication methods.

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

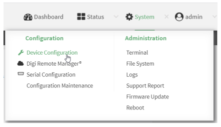
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Delete an authentication method

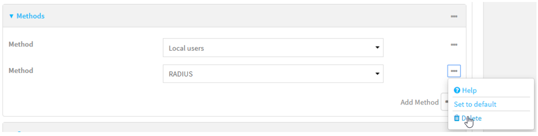
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.

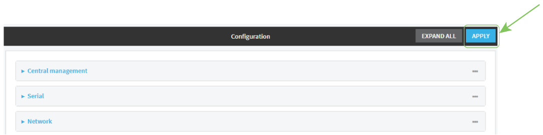
3. Click **Authentication** > **Methods**.
4. Click the menu icon (**...**) next to the method and select **Delete**.


Methods

Method: Local users

Method: RADIUS

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Use the **show auth method** command to determine the index number of the authentication method to be deleted:

```
(config)> show auth method
0 local
1 radius
2 tacacs+
(config)>
```

4. Delete the appropriate authentication method:

```
(config)> del auth method n
```

   Where *n* is index number of the authentication method to be deleted. For example, to delete the TACACS+ authentication method as displayed by the example **show** command, above:

```
(config)> del auth method 2
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

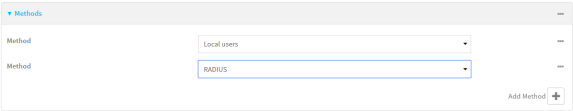
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Rearrange the position of authentication methods

WebUI

Authentication methods are reordered by changing the method type in the **Method** drop-down for each authentication method to match the appropriate order.

For example, the following configuration has **Local users** as the first method, and **RADIUS** as the second.

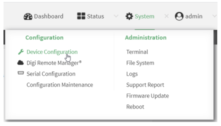


Method: Local users

Method: RADIUS

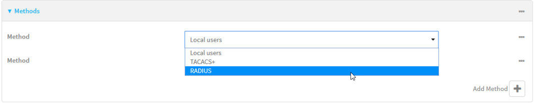
To reorder these so that **RADIUS** is first and **Local users** is second:

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

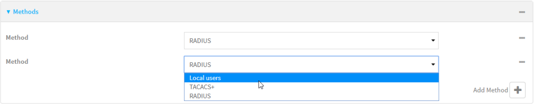


The **Configuration** window is displayed.

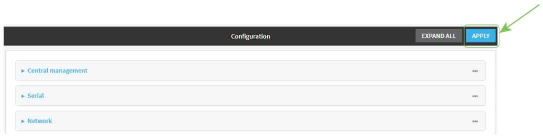
3. Click to expand the first **Method**.
4. In the **Method** drop-down, select **RADIUS**.



5. Click to expand the second **Method**.
6. In the **Method** drop-down, select **Local users**.



7. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Use the **show** command to display current configuration:

```
(config)> show auth method
0 local
1 radius
(config)>
```

4. Use the **move** command to rearrange the methods:

```
(config)> move auth method 1 0
(config)>
```

5. Use the **show** command again to verify the change:

```
(config)> show auth method
0 radius
1 local
(config)>
```

6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Authentication groups

Authentication groups are used to assign access rights to AnywhereUSB Plus users. Three types of access rights can be assigned:

- **Admin access**: Users with Admin access can be configured to have either:
  - The ability to manage the AnywhereUSB Plus device by using the WebUI or the Admin CLI.
  - Read-only access to the WebUI and Admin CLI.
- **Shell access**: Users with Shell access have the ability to access the shell when logging into the AnywhereUSB Plus via ssh or the serial console.

  Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.
- **Serial access**: Users with Serial access have the ability to log into the AnywhereUSB Plus device by using the serial console.

Preconfigured authentication groups

The AnywhereUSB Plus device has two preconfigured authentication groups:

- The **admin** group is configured by default to have full **Admin access** and **Shell access**.

  Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.
- The **serial** group is configured by default to have **Serial access**.

The preconfigured authentication groups cannot be deleted, but the access rights defined for the group are configurable.

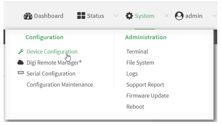
This section contains the following topics:

Change the access rights for a predefined group

By default, two authentication groups are predefined: **admin** and **serial**. To change the access rights of the predefined groups:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

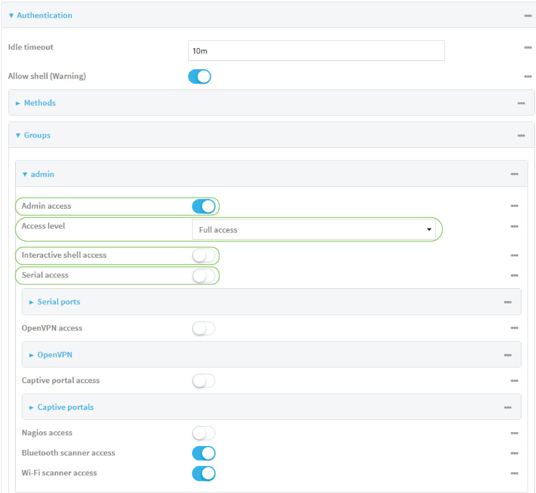
   The **Configuration** window is displayed.
3. Click **Authentication** > **Groups**.
4. Click the authentication group to be changed, either **admin** or **serial**, to expand its configuration node.
5. Click the box next to the following options, as appropriate, to enable or disable access rights for each:
   - **Admin access**

     For groups assigned Admin access, you can also determine whether the **Access level** should be **Full access** or **Read-only access**.

     - **Full access** provides users of this group with the ability to manage the AnywhereUSB Plus device by using the WebUI or the Admin CLI.
     - **Read-only access** provides users of this group with read-only access to the WebUI and Admin CLI.

     The default is **Full access**.
   - **Interactive shell access**

     Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.
   - **Serial access**



▾ Authentication

Idle timeout: 10m

Allow shell (Warning)

▸ Methods

▾ Groups

▾ admin

Admin access

Access level: Full access

Interactive shell access

Serial access

▸ Serial ports

OpenVPN access

▸ OpenVPN

Captive portal access

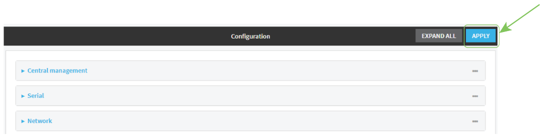
▸ Captive portals

Nagios access

Bluetooth scanner access

Wi-Fi scanner access

6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Enable or disable access rights for the group. For example:

   - Admin access:

     - To set the access level for Admin access of the **admin** group:

       ```
       (config)> auth group admin acl admin level value
       (config)>
       ```

       where *value* is either:

       - **full**: provides users of this group with the ability to manage the AnywhereUSB Plus device by using the WebUI or the Admin CLI.
       - **read-only**: provides users of this group with read-only access to the WebUI and Admin CLI.

       The default is **full**.

        - To disable Admin access for the **admin** group:

```
(config)> auth group admin acl admin enable false
(config)>
```

    - Shell access:
        - To enable Shell access for the **serial** group:

```
(config)> auth group serial acl shell enable true
(config)>
```

Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.

    - Serial access:
        - To enable Serial access for the **admin** group:

```
(config)> auth group admin acl serial enable true
(config)>
```

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Add an authentication group

Required configuration items

- The access rights to be assigned to users that are assigned to this group.

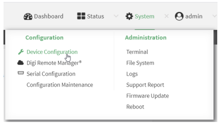
Additional configuration items

- Access rights to OpenVPN tunnels, and the tunnels to which they have access.
- Access rights to captive portals, and the portals to which they have access.
- Access rights to query the device for Nagios monitoring.

To add an authentication group:

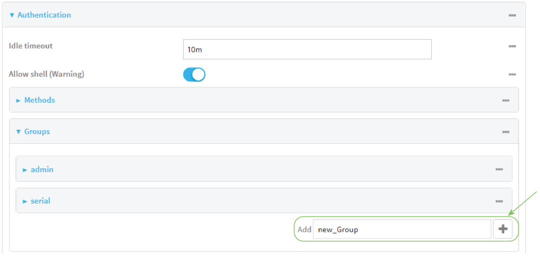
☰ **WebUI**

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Authentication** > **Groups**.
4. For **Add**, type a name for the group and click **+**.

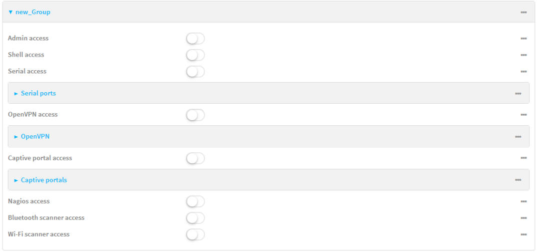


Idle timeout 10m

Allow shell (Warning) ●

Add new_Group +

The group configuration window is displayed.



Admin access ○

Shell access ○

Serial access ○

OpenVPN access ○

Captive portal access ○

Nagios access ○

Bluetooth scanner access ○

Wi-Fi scanner access ○

5. Click the following options, as appropriate, to enable or disable access rights for each:
   - **Admin access**

     For groups assigned Admin access, you can also determine whether the **Access level** should be **Full access** or **Read-only access**.

     where *value* is either:

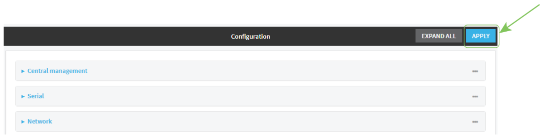
     - **Full access full**: provides users of this group with the ability to manage the AnywhereUSB Plus device by using the WebUI or the Admin CLI.
     - **Read-only access read-only**: provides users of this group with read-only access to the WebUI and Admin CLI.

     The default is **Full access full**.
   - **Shell access**

     Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.
   - **Serial access**
6. (Optional) Configure OpenVPN access. See for further information.
7. (Optional) Configure captive portal access:

   a. Enable captive portal access rights for users of this group by checking the box next to **Captive portal access**.
   b. Click **Captive portals** to expand the **Captive portal** node.
   c. For **Add Captive portal**, click ✚.
   d. In the **Captive portal** dropdown, select a captive portal to which users of this group will have access.
   e. Click ✚ again to add additional captive portals.

8. (Optional) Enable users that belong to this group to query the device for Nagios monitoring by checking the box next to **Nagios access**.
9. (Optional) Enable users that belong to this group to access the Bluetooth scanning service by checking the box next to **Bluetooth scanner access**.
10. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Use the **add auth group** command to add a new authentication. For example, to add a group named **test**:

```
(config)> add auth group test
(config auth group test)>
```

4. Enable access rights for the group:
   - Admin access:

   ```
   (config auth group test)> acl admin enable true
   (config)>
   ```

   - Set the access level for Admin access:

   ```
   (config)> auth group admin acl admin level value
   (config)>
   ```

   where *value* is either:
     - **full**: provides users of this group with the ability to manage the AnywhereUSB Plus device by using the WebUI or the Admin CLI.

    - **read-only**: provides users of this group with read-only access to the WebUI and Admin CLI.

    The default is **full**.

- Shell access:

  ```
  (config auth group test)> acl shell enable true
  (config)>
  ```

  Shell access is not available if the **Allow shell** parameter has been disabled. See Disable shell access for more information about the **Allow shell** parameter.

- Serial access:

  ```
  (config auth group test)> acl serial enable true
  (config)>
  ```

5. (Optional) Configure captive portal access:

   a. Return to the config prompt by typing three periods (**...**):

   ```
   (config auth group test)> ...
   (config)>
   ```

   b. Enable captive portal access rights for users of this group:

   ```
   (config)> auth group test acl portal enable true
   (config)>
   ```

   c. Add a captive portal to which users of this group will have access:

      i. Determine available portals:

      ```
      (config)> show firewall portal
      portal1
              auth none
              enable true
              http redirect
              no interface
              no message
              no redirect_url
              no terms
              timeout 24h
              no title
      (config)>
      ```

      ii. Add a captive portal:

      ```
      (config)> add auth group test acl portal portals end portal1
      (config)>
      ```

6. (Optional) Configure Nagios monitoring:

   ```
   (config)> auth group test acl nagios enable true
   (config)>
   ```

7. (Optional) Enable users that belong to this group to access the Bluetooth scanning service:

```
(config)> auth group test acl bluetooth_scanner enable true
(config)>
```

8. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

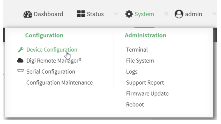
Delete an authentication group

By default, the AnywhereUSB Plus device has two preconfigured authentication groups: **admin** and **serial**. These groups cannot be deleted.

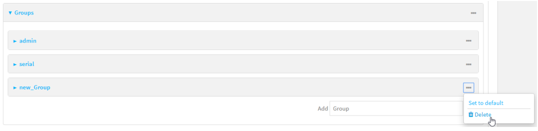
To delete an authentication group that you have created:

WebUI

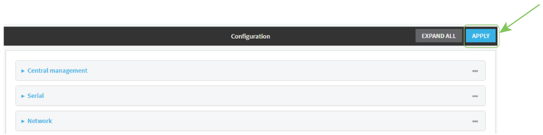
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Authentication** > **Groups**.
4. Click the menu icon **(...)** next to the group to be deleted and select **Delete**.



5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. At the config prompt, type:

   ```
   (config)> del auth group groupname
   ```

4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Local users

Local users are authenticated on the device without using an external authentication mechanism such as TACACS+ or RADIUS. Local user authentication is enabled by default, with one preconfiged default user.

Default user

At manufacturing time, each AnywhereUSB Plus device comes with a default user configured as follows:

- Username: **admin**.
- Password: The default password is displayed on the label on the bottom of the device.

Note The default password is a unique password for the device, and is the most critical security feature for the device. If you reset the device to factory defaults, you must log in using the default user and password, and you should immediately change the password to a custom password. Before deploying or mounting the AnywhereUSB Plus device, record the default password, so you have the information available when you need it even if you cannot physically access the label on the bottom of the device.

The default **admin** user is preconfigured with both Admin and Serial access. You can configure the **admin** user account to fit with the needs of your environment.

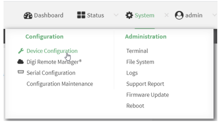
This section contains the following topics:

Change a local user's password

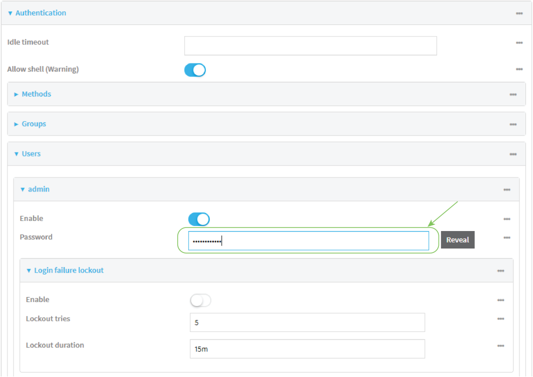
To change a user's password:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

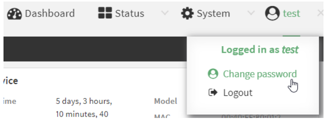
   

   The **Configuration** window is displayed.
3. Click **Authentication** > **Users**.
4. Click the username to expand the user's configuration node.
5. For **Password**, enter the new password. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.

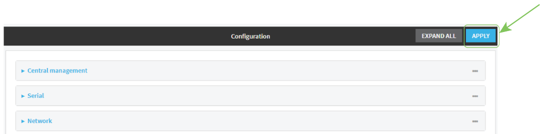
   

   Idle timeout
   Allow shell (Warning)
   Enable
   Password ••••••••••••
   Enable
   Lockout tries 5
   Lockout duration 15m

   You can also change the password for the active user by clicking the user name in the menu bar:

   The active user must have full Admin access rights to be able to change the password.
6. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. At the config prompt, type:

   ```
   (config)> auth user username password pwd
   ```

   Where:

   - *username* is the name of the user.
   - *pwd* is the new password for the user. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure a local user

Required configuration items

- A username.
- A password. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character. For security reasons, passwords are stored in hash form. There is no way to get or display passwords in clear-text form, although prior to saving the configuration, the password can be shown by clicking **Reveal**.
- The authentication group or groups from which the user will inherit access rights. See Authentication groups for information about configuring groups.

Additional configuration items

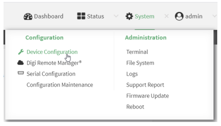
- The number of unsuccessful login attempts before the user is locked out of the system.
- The amount of time that the user is locked out of the system after the specified number of unsuccessful login attempts.
- An optional public ssh key, to authenticate the user when using passwordless SSH login.

- Two-factor authentication information for user login over SSH and the serial console:
  - The verification type for two-factor authentication: Either time-based or counter-based.
  - The security key.
  - Whether to allow passcode reuse (time based verification only).
  - The passcode refresh interval (time based verification only).
  - The valid code window size.
  - The login limit.
  - The login limit period.
  - One-time use eight-digit emergency scratch codes.

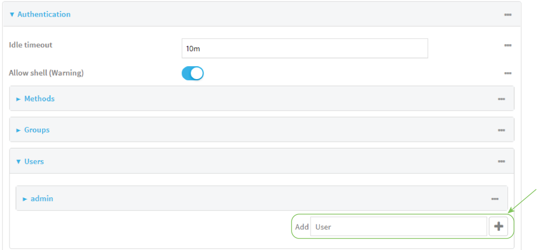
To configure a local user:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



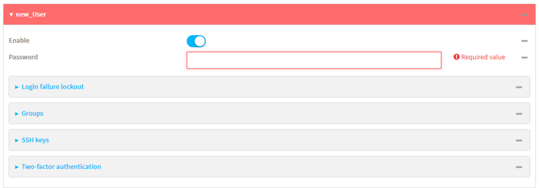
   The **Configuration** window is displayed.
3. Click **Authentication** > **Users**.
4. In **Add User**, type a name for the user and click ✚.



Idle timeout 10m

Allow shell (Warning)

   The user configuration window is displayed.



Enable

Password

Required value

   The user is enabled by default. To disable, click to toggle off **Enable**.
5. Enter a password for the user. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.
6. Click to expand **Login failure lockout**.

   The login failure lockout feature is enabled by default. To disable, click to toggle off **Enable**.

   a. For **Lockout tries**, type the number of unsuccessful login attempts before the user is locked out of the device. The default is **5**.
   b. For **Lockout duration**, type the amount of time that the user is locked out after the number of unsuccessful login attempts defined in **Lockout tries**.

      Allowed values are any number of minutes, or seconds, and take the format ***number***{**m**|**s**}.

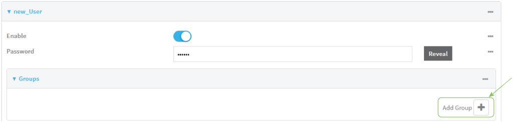
      For example, to set **Lockout duration** to ten minutes, enter **10m** or **600s**.

      The minimum value is 1 second, and the maximum is 15 minutes. The default is 15 minutes.

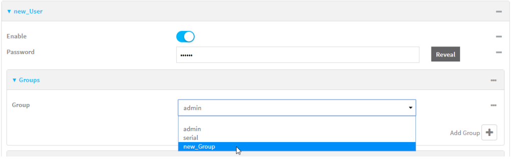
7. Add groups for the user.

   Groups define user access rights. See Authentication groups for information about configuring groups.

   a. Click to expand **Groups**.
   b. For **Add Group**, click ✚.



   c. For **Group**, select an appropriate group.

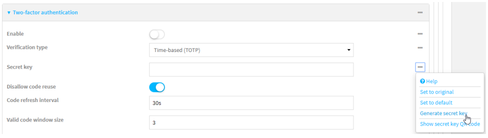


Note Every user must be configured with at least one group. You can add multiple groups to a user by clicking **Add** again and selecting the next group.

8. (Optional) Add SSH keys for the user to use passwordless SSH login:
   a. Click **SSH keys**.
   b. In **Add SSH key**, paste or type a public encryption key that this user can use for passwordless SSH login and click ✚.
9. (Optional) Configure two-factor authentication for SSH and serial console login:
   a. Click **Two-factor authentication**.
   b. Check **Enable** to enable two-factor authentication for this user.
   c. Select the **Verification type**:
      - **Time-based (TOTP)**: Time-based One-Time Password (TOTP) authentication uses the current time to generate a one-time password.
      - **Counter-based (HOTP)**: HMAC-based One-Time Password (HOTP) uses a counter to validate a one-time password.

d. Generate a **Secret key**:

   i. Click **...** next to the field label and select **Generate secret key**.



Two-factor authentication

Enable

Verification type: Time-based (TOTP)

Secret key

Disallow code reuse

Code refresh interval: 30s

Valid code window size: 3

   ii. To display the QR code for the secret key, click **...** next to the field label and select **Show secret key QR code**.

   iii. Copy the secret key, or scan or copy the QR code, for use with an application or mobile device to generate passcodes.

> **Note** To copy the QR code, right-click the QR code and select your browser's save image functionality.

e. For time-based verification only, select **Disallow code reuse** to prevent a code from being used more than once during the time that it is valid.

f. For time-based verification only, in **Code refresh interval**, type the amount of time that a code will remain valid.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}. For example, to set **Code refresh interval** to ten minutes, enter **10m** or **600s**.

g. In **Valid code window size**, type the allowed number of concurrently valid codes. In cases where TOTP is being used, increasing the **Valid code window size** may be necessary when the clocks used by the server and client are not synchronized.

h. For **Login limit**, type the number of times that the user is allowed to attempt to log in during the **Login limit period**. Set **Login limit** to **0** to allow an unlimited number of login attempts during the **Login limit period**.

i. For **Login limit period**, type the amount of time that the user is allowed to attempt to log in.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}. For example, to set **Login limit period** to ten minutes, enter **10m** or **600s**.

j. Scratch codes are emergency codes that may be used once, at any time. To add a scratch code:

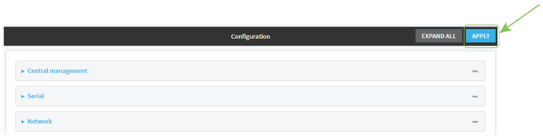
   i. Click **Scratch codes**.

   ii. For **Add Code**, click **✚**.

   iii. For **Code**, enter the scratch code. The code must be eight digits, with a minimum of 10000000.

   iv. Click **✚** again to add additional scratch codes.

10. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add a user. For example, to create a user named **new_user**:

   ```
   (config)> add auth user new_user
   (config auth user new_user)>
   ```

   The user is enabled by default. To disable the user, type:

   ```
   (config auth user new_user)> enable false
   (config auth user new_user)>
   ```
4. Set the user's password. The password must be at least ten characters long and must contain at least one uppercase letter, one lowercase letter, one number, and one special character.

   ```
   (config auth user new_user> password pwd
   (config auth user new_user)>
   ```
5. Configure login failure lockout settings:

   The login failure lockout feature is enabled by default. To disable:

   ```
   (config auth user new_user> lockout enable false
   (config auth user new_user)>
   ```

   a. Set the number of unsuccessful login attempts before the user is locked out of the device.

      where *value* is any integer. The minimum value is **1**, and the default value is **5**.

   b. Set the amount of time that the user is locked out after the number of unsuccessful login attempts defined in **lockout tries**:

      ```
      (config auth user new_user> lockout duration value
      (config auth user new_user)>
      ```

      where *value* is any number of minutes, or seconds, and takes the format ***number*{m|s}**.

      For example, to set **duration** to ten minutes, enter either **10m** or **600s**:

      ```
      (config auth user new_user)> lockout duration 600s
      (config auth user new_user)>
      ```

The minimum value is 1 second, and the maximum is 15 minutes. The default is 15 minutes.

6. Add groups for the user.

   Groups define user access rights. See Authentication groups for information about configuring groups.

   a. Add a group to the user. For example, to add the admin group to the user:

   ```
   (config auth user new_user> add group end admin
   (config auth user new_user)>
   ```

   Note Every user must be configured with at least one group.

   b. (Optional) Add additional groups by repeating the add group command:

   ```
   (config auth user new_user> add group end serial
   (config auth user new_user)>
   ```

   To remove a group from a user:

   a. Use the **show** command to determine the index number of the group to be deleted:

   ```
   (config auth user new_user> show group
   0 admin
   1 serial
   (config auth user new_user>
   ```

   b. Type the following:

   ```
   (config auth user new_user)> del group n
   (config auth user new_user)>
   ```

   Where *n* is index number of the authentication method to be deleted. For example, to delete the serial group as displayed by the example **show** command, above:

   ```
   (config auth user new_user)> del group 1
   (config auth user new_user)>
   ```

7. (Optional) Add SSH keys for the user to use passwordless SSH login:

   a. Change to the user's ssh_key node:

   ```
   (config auth user new_user)> ssh_key
   (config auth user new_user ssh_key)>
   ```

   b. Add the key by using the ssh_key command and pasting or typing a public encryption key that this user can use for passwordless SSH login:

   ```
   (config auth user new_user ssh_key)> ssh_key key
   (config auth user new_user ssh_key)>
   ```

8. (Optional) Configure two-factor authentication for SSH and serial console login:

a. Change to the user's two-factor authentication node:

```
(config auth user new_user)> 2fa
(config auth user new_user 2fa)>
```

b. Enable two-factor authentication for this user:

```
(config auth user new_user 2fa)> enable true
(config auth user new_user 2fa)>
```

c. Configure the verification type. Allowed values are:

- **totp**: Time-based One-Time Password (TOTP) authentication uses the current time to generate a one-time password.
- **hotp**: HMAC-based One-Time Password (HOTP) uses a counter to validate a one-time password.

The default value is **totp**.

```
(config auth user new_user 2fa)> type totp
(config auth user new_user 2fa)>
```

d. Add a secret key:

```
(config auth user new_user 2fa)> secret key
(config auth user new_user 2fa)>
```

This key should be used by an application or mobile device to generate passcodes.

e. For time-based verification only, enable **disallow_reuse** to prevent a code from being used more than once during the time that it is valid.

```
(config auth user new_user 2fa)> disallow_reuse true
(config auth user new_user 2fa)>
```

f. For time-based verification only, configure the code refresh interval. This is the amount of time that a code will remain valid.

```
(config auth user new_user 2fa)> refresh_interval value
(config auth user new_user 2fa)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **refresh_interval** to ten minutes, enter either **10m** or **600s**:

```
(config auth user name 2fa)> refresh_interval 600s
(config auth user name 2fa)>
```

The default is **30s**.

g. Configure the valid code window size. This represents the allowed number of concurrently valid codes. In cases where TOTP is being used, increasing the valid code window size may be necessary when the clocks used by the server and client are not synchronized.

```
(config auth user new_user 2fa)> window_size 3
(config auth user new_user 2fa)>
```

h. Configure the login limit. This represents the number of times that the user is allowed to attempt to log in during the Login limit period. Set to 0 to allow an unlimited number of login attempts during the Login limit period

```
(config auth user new_user 2fa)> login_limit 3
(config auth user new_user 2fa)>
```

i. Configure the login limit period. This is the amount of time that the user is allowed to attempt to log in.

```
(config auth user new_user 2fa)> login_limit_period value
(config auth user new_user 2fa)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

For example, to set **login_limit_period** to ten minutes, enter either **10m** or **600s**:

```
(config auth user name 2fa)> login_limit_period 600s
(config auth user name 2fa)>
```

The default is **30s**.

j. Scratch codes are emergency codes that may be used once, at any time. To add a scratch code:

   i. Change to the user's scratch code node:

```
(config auth user new_user 2fa)> scratch_code
(config auth user new_user 2fa scratch_code)>
```

   ii. Add a scratch code:

```
(config auth user new_user 2fa scratch_code)> add end code
(config auth user new_user 2fa scratch_code)>
```

   Where code is an digit number, with a minimum of 10000000.

   iii. To add additional scratch codes, use the **add end *code*** command again.

9. Save the configuration and apply the change:

```
(config auth user new 2fa scratch_code)> save
Configuration saved.
>
```

10. Type **exit** to exit the Admin CLI.

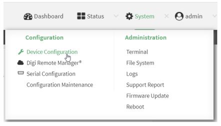
    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Delete a local user

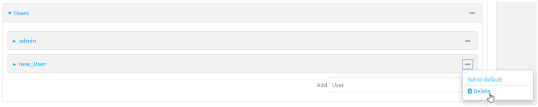
To delete a user from your AnywhereUSB Plus:

☰ WebUI

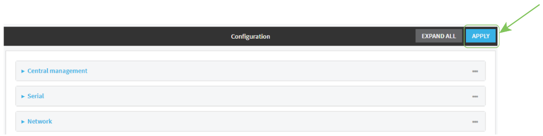
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Authentication** > **Users**.
4. Click the menu icon (**...**) next to the name of the user to be deleted and select **Delete**.

5. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. At the config prompt, type:

   ```
   (config)> del auth user username
   ```

4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Terminal Access Controller Access-Control System Plus (TACACS+)

Your AnywhereUSB Plus device supports Terminal Access Controller Access-Control System Plus (TACACS+), a networking protocol that provides centralized authentication and authorization management for users who connect to the device. With TACACS+ support, the AnywhereUSB Plus device acts as a TACACS+ client, which sends user credentials and connection parameters to a TACACS+ server over TCP. The TACACS+ server then authenticates the TACACS+ client requests and sends back a response message to the device.

When you are using TACACS+ authentication, you can have both local users and TACACS+ users able to log in to the device. To use TACACS+ authentication, you must set up a TACACS+ server that is accessible by the AnywhereUSB Plus device prior to configuration. The process of setting up a TACACS+ server varies by the server environment.

This section contains the following topics:

TACACS+ user configuration

When configured to use TACACS+ support, the AnywhereUSB Plus device uses a remote TACACS+ server for user authentication (password verification) and authorization (assigning the access level of the user). Additional TACACS+ servers can be configured as backup servers for user authentication.

This section outlines how to configure a TACACS+ server to be used for user authentication on your AnywhereUSB Plus device.

Example TACACS+ configuration

With TACACS+, users are defined in the server configuration file. On Ubuntu, the default location and filename for the server configuration file is **/etc/tacacs+/tac_plus.conf**.

Note TACACS+ configuration, including filenames and locations, may vary depending on your platform and installation. This example assumes a Ubuntu installation.

To define users:

1. Open the TACACS+ server configuration file in a text editor. For example:

```
$ sudo gedit /etc/tacacs+/tac_plus.conf
```

2. Add users to the file using the following format. This example will create two users, one with admin and serial access, and one with only serial access.

```
user = user1 {
       name ="User1 for AnywhereUSB Plus"
       pap = cleartext password1
       service = system {
         groupname = admin,serial
       }
}
user = user2 {
       name ="User2 for AnywhereUSB Plus"
       pap = cleartext password2
       service = system {
         groupname = serial
       }
}
```

The **groupname** attribute is optional. If used, the value must correspond to authentication groups configured on your AnywhereUSB Plus. Alternatively, if the user is also configured as a local user on the AnywhereUSB Plus device and the LDAP server authenticates the user but does not return any groups, the local configuration determines the list of groups. See Authentication groups for more information about authentication groups. The **groupname** attribute can contain one group or multiple groups in a comma-separated list.

3. Save and close the file.
4. Verify that your changes did not introduce any syntax errors:

```
$ sudo tac_plus -C /etc/tacacs+/tac_plus.conf -P
```

If successful, this command will echo the configuration file to standard out. If the command encounters any syntax errors, a message similar to this will display:

```
Error: Unrecognised token on line 1
```

5. Restart the TACACS+ server:

```
$ sudo /etc/init.d/tacacs_plus restart
```

TACACS+ server failover and fallback to local authentication

In addition to the primary TACACS+ server, you can also configure your AnywhereUSB Plus device to use backup TACACS+ servers. Backup TACACS+ servers are used for authentication requests when the primary TACACS+ server is unavailable.

Falling back to local authentication

With user authentication methods, you can configure your AnywhereUSB Plus device to use multiple types of authentication. For example, you can configure both TACACS+ authentication and local authentication, so that local authentication can be used as a fallback mechanism if the primary and backup TACACS+ servers are unavailable. Additionally, users who are configured locally but are not configured on the TACACS+ server are still able to log into the device. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned; therefore if you want to ensure that users are authenticated first through the TACACS+ server, and only authenticated locally if the TACACS+ server is unavailable or if the user is not defined on the TACACS+ server, then you should list the TACACS+ authentication method prior to the Local users authentication method.

See User authentication methods for more information about authentication methods.

If the TACACS+ servers are unavailable and the AnywhereUSB Plus device falls back to local authentication, only users defined locally on the device are able to log in. TACACS+ users cannot log in until the TACACS+ servers are brought back online.

Configure your AnywhereUSB Plus device to use a TACACS+ server

This section describes how to configure a AnywhereUSB Plus device to use a TACACS+ server for authentication and authorization.

Required configuration items

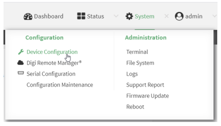
- Define the TACACS+ server IP address or domain name.
- Define the TACACS+ server shared secret.
- The group attribute configured in the TACACS+ server configuration.
- The service field configured in the TACACS+ server configuration.
- Add TACACS+ as an authentication method for your AnywhereUSB Plus device.

Additional configuration items

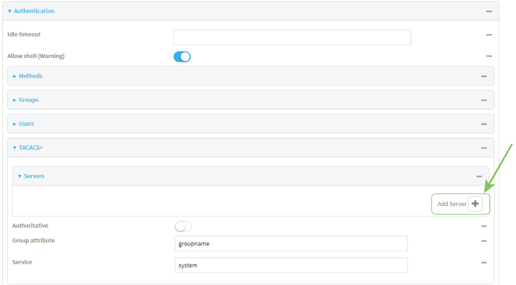
- Whether other user authentication methods should be used in addition to the TACACS+ server, or if the TACACS+ server should be considered the authoritative login method.
- The TACACS+ server port. It is configured to 49 by default.
- Add additional TACACS+ servers in case the first TACACS+ server is unavailable.

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Authentication** > **TACACS+** > **Servers**.
4. Add TACACS+ servers:
   a. For **Add server**, click ✚.

   Group attribute: groupname

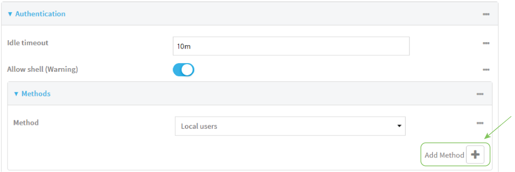
   Service: system

   b. For **Hostname**, type the hostname or IP address of the TACACS+ server.
   c. (Optional) Change the default **Port** setting to the appropriate port. Normally this should be left at the default setting of port 49.
   d. For **Secret**, type the TACACS+ server's shared secret. This is configured in the key parameter of the TACACS+ server's tac_plus.conf file, for example:

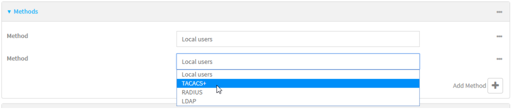
   ```
   key = testing123
   ```

   e. (Optional) Click ✚ again to add additional TACACS+ servers.
5. (Optional) Enable **Authoritative** to prevent other authentication methods from being used if TACACS+ authentication fails. Other authentication methods will only be used if the TACACS+ server is unavailable.
6. (Optional) For **Group attribute**, type the name of the attribute used in the TACACS+ server's configuration to identify the AnywhereUSB Plus authentication group or groups that the user is a member of. For example, in TACACS+ user configuration, the group attribute in the sample tac_plus.conf file is **groupname**, which is also the default setting in the AnywhereUSB Plus configuration.
7. (Optional) For **Service**, type the value of the **service** attribute in the the TACACS+ server's configuration. For example, in TACACS+ user configuration, the value of the **service** attribute in the sample tac_plus.conf file is **system**, which is also the default setting in the AnywhereUSB Plus configuration.

8. Add TACACS+ to the authentication methods:
   a. Click **Authentication** > **Methods**.
   b. For **Add method**, click ✚.

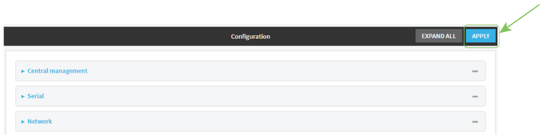


   c. Select **TACACS+** for the new method from the **Method** drop-down.



   Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. See Rearrange the position of authentication methods for information about rearranging the position of the methods in the list.

9. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. (Optional) Prevent other authentication methods from being used if TACACS+ authentication fails. Other authentication methods will only be used if the TACACS+ server is unavailable.

```
(config)> auth tacacs+ authoritative true
(config)>
```

4. (Optional) Configure the group_attribute. This is the name of the attribute used in the TACACS+ server's configuration to identify the AnywhereUSB Plus authentication group or groups that the user is a member of. For example, in TACACS+ user configuration, the group attribute in the sample tac_plus.conf file is **groupname**, which is also the default setting for the group_attribute in the AnywhereUSB Plus configuration.

```
(config)> auth tacacs+ group_attribute attribute-name
(config)>
```

5. (Optional) Configure the type of service. This is the value of the **service** attribute in the the TACACS+ server's configuration. For example, in TACACS+ user configuration, the value of the **service** attribute in the sample tac_plus.conf file is **system**, which is also the default setting in the AnywhereUSB Plus configuration.

   ```
   (config)> auth tacacs+ service service-name
   (config)>
   ```

6. Add a TACACS+ server:
   1. Add the server:

      ```
      (config)> add auth tacacs+ server end
      (config auth tacacs+ server 0)>
      ```

   2. Enter the TACACS+ server's IP address or hostname:

      ```
      (config auth tacacs+ server 0)> hostname hostname|ip-address
      (config auth tacacs+ server 0)>
      ```

   3. (Optional) Change the default port setting to the appropriate port:

      ```
      (config auth tacacs+ server 0)> port port
      (config auth tacacs+ server 0)>
      ```

   4. (Optional) Repeat the above steps to add additional TACACS+ servers.

7. Add TACACS+ to the authentication methods. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. This example will add TACACS+ to the end of the list. See User authentication methods for information about adding methods to the beginning or middle of the list.

   ```
   (config)> add auth method end tacacs+
   (config)>
   ```

8. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Remote Authentication Dial-In User Service (RADIUS)

Your AnywhereUSB Plus device supports Remote Authentication Dial-In User Service (RADIUS), a networking protocol that provides centralized authentication and authorization management for users who connect to the device. With RADIUS support, the AnywhereUSB Plus device acts as a RADIUS client, which sends user credentials and connection parameters to a RADIUS server over UDP. The RADIUS server then authenticates the RADIUS client requests and sends back a response message to the device.

When you are using RADIUS authentication, you can have both local users and RADIUS users able to log in to the device. To use RADIUS authentication, you must set up a RADIUS server that is accessible by the AnywhereUSB Plus device prior to configuration. The process of setting up a RADIUS server varies by the server environment. An example of a RADIUS server is FreeRADIUS.

This section contains the following topics:

RADIUS user configuration

When configured to use RADIUS support, the AnywhereUSB Plus device uses a remote RADIUS server for user authentication (password verification) and authorization (assigning the access level of the user). Additional RADIUS servers can be configured as backup servers for user authentication.

This section outlines how to configure a RADIUS server to be used for user authentication on your AnywhereUSB Plus device.

Example FreeRADIUS configuration

With FreeRADIUS, users are defined in the **users** file in your FreeRADIUS installation. To define users:

1. Open the FreeRadius user file in a text editor. For example:

   ```
   $ sudo gedit /etc/freeradius/3.0/users
   ```

2. Add users to the file using the following format:

   ```
   user1 Cleartext-Password := "user1"
          Unix-FTP-Group-Names := "admin"

   user2 Cleartext-Password := "user2"
          Unix-FTP-Group-Names := "serial"
   ```

   The **Unix-FTP-Group-Names** attribute is optional. If used, the value must correspond to authentication groups configured on your AnywhereUSB Plus. Alternatively, if the user is also configured as a local user on the AnywhereUSB Plus device and the RADIUS server authenticates the user but does not return any groups, the local configuration determines the list of groups. See Authentication groups for more information about authentication groups. The **Unix-FTP-Group-Names** attribute can contain one group or multiple groups in a comma-separated list.

3. Save and close the file.
4. Verify that your changes did not introduce any syntax errors:

   ```
   $ sudo freeradius -CX
   ```

   This should return a message that completes similar to:

   ```
   ...
   Configuration appears to be OK
   ```

5. Restart the FreeRADIUS server:

   ```
   $ sudo /etc/init.d/freeradius restart
   ```

RADIUS server failover and fallback to local configuration

In addition to the primary RADIUS server, you can also configure your AnywhereUSB Plus device to use backup RADIUS servers. Backup RADIUS servers are used for authentication requests when the primary RADIUS server is unavailable.

Falling back to local authentication

With user authentication methods, you can configure your AnywhereUSB Plus device to use multiple types of authentication. For example, you can configure both RADIUS authentication and local authentication, so that local authentication can be used as a fallback mechanism if the primary and backup RADIUS servers are unavailable. Additionally, users who are configured locally but are not configured on the RADIUS server are still able to log into the device. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned; therefore if you want to ensure that users are authenticated first through the RADIUS server, and only authenticated locally if the RADIUS server is unavailable or if the user is not defined on the RADIUS server, then you should list the RADIUS authentication method prior to the Local users authentication method.

See User authentication methods for more information about authentication methods.

If the RADIUS servers are unavailable and the AnywhereUSB Plus device falls back to local authentication, only users defined locally on the device are able to log in. RADIUS users cannot log in until the RADIUS servers are brought back online.

Configure your AnywhereUSB Plus device to use a RADIUS server

This section describes how to configure a AnywhereUSB Plus device to use a RADIUS server for authentication and authorization.

Required configuration items

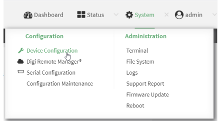
- Define the RADIUS server IP address or domain name.
- Define the RADIUS server shared secret.
- Add RADIUS as an authentication method for your AnywhereUSB Plus device.

Additional configuration items

- Whether other user authentication methods should be used in addition to the RADIUS server, or if the RADIUS server should be considered the authoritative login method.
- The RADIUS server port. It is configured to 1812 by default.
- Add additional RADIUS servers in case the first RADIUS server is unavailable.
- The server NAS ID. If left blank, the default value is used:
  - If you are access the AnywhereUSB Plus device by using the WebUI, the default value is for NAS ID is **httpd**.
  - If you are access the AnywhereUSB Plus device by using ssh, the default value is **sshd**.
- Time in seconds before the request to the server times out. The default is 3 seconds and the maximum possible value is 60 seconds.
- Enable additional debug messages from the RADIUS client.

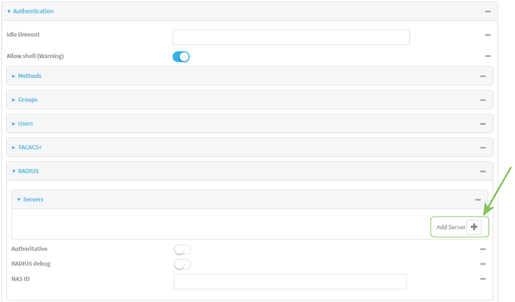
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Authentication** > **RADIUS** > **Servers**.

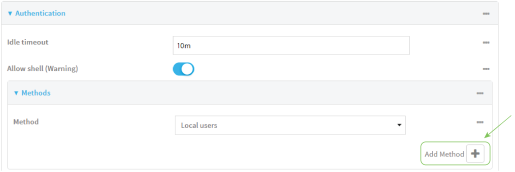
4. Add RADIUS servers:
   a. For **Add server**, click ✚.



   b. For **Hostname**, type the hostname or IP address of the RADIUS server.
   c. (Optional) Change the default **Port** setting to the appropriate port. Normally this should be left at the default setting of port 1812.
   d. For **Secret**, type the RADIUS server's shared secret. This is configured in the secret parameter of the RADIUS server's client.conf file, for example:

```
secret=testing123
```

   e. For **Timeout**, type or select the amount of time in seconds to wait for the RADIUS server to respond. Allowed value is any integer from **3** to **60**. The default value is **3**.
   f. (Optional) Click ✚ again to add additional RADIUS servers.
5. (Optional) Enable **Authoritative** to prevent other authentication methods from being used if RADIUS authentication fails. Other authentication methods will only be used if the RADIUS server is unavailable.
6. (Optional) Click **RADIUS debug** to enable additional debug messages from the RADIUS client.
7. (Optional) For **NAS ID**, type the unique identifier for this network access server (NAS). You can use the fully-qualified domain name of the NAS or any arbitrary string. If not set, the default value is used:
   - If you are accessing the AnywhereUSB Plus device by using the WebUI, the default value is for NAS ID is **httpd**.
   - If you are accessing the AnywhereUSB Plus device by using ssh, the default value is **sshd**.
8. Add RADIUS to the authentication methods:
   a. Click **Authentication** > **Methods**.
   b. For **Add method**, click ✚.

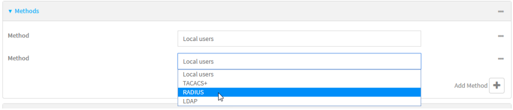


Idle timeout 10m

Allow shell (Warning)

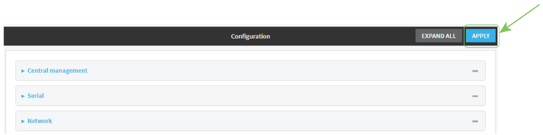
Method Local users

c. Select **RADIUS** for the new method from the **Method** drop-down.



Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. See Rearrange the position of authentication methods for information about rearranging the position of the methods in the list.

9. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. (Optional) Prevent other authentication methods from being used if RADIUS authentication fails. Other authentication methods will only be used if the RADIUS server is unavailable.

```
(config)> auth radius authoritative true
(config)>
```

4. (Optional) Enable debug messages from the RADIUS client:

```
(config)> auth radius debug true
(config)>
```

5. (Optional) Configure the NAS ID. This is a unique identifier for this network access server (NAS). You can use the fully-qualified domain name of the NAS or any arbitrary string. If not set, the default value is used:
   - If you are accessing the AnywhereUSB Plus device by using the WebUI, the default value is for NAS ID is **httpd**.
   - If you are accessing the AnywhereUSB Plus device by using ssh, the default value is **sshd**.

```
(config)> auth radius nas_id id
(config)>
```

6. Add a RADIUS server:
   a. Add the server:

      ```
      (config)> add auth radius server end
      (config auth radius server 0)>
      ```

   b. Enter the RADIUS server's IP address or hostname:

      ```
      (config auth radius server 0)> hostname hostname|ip-address
      (config auth radius server 0)>
      ```

   c. (Optional) Change the default port setting to the appropriate port:

      ```
      (config auth radius server 0)> port port
      (config auth radius server 0)>
      ```

   d. Configure the amount of time in seconds to wait for the RADIUS server to respond. Allowed value is any integer from **3** to **60**. The default value is **3**.

      ```
      (config auth radius server 0)> timeout value
      (config auth radius server 0)>
      ```

   e. (Optional) Repeat the above steps to add additional RADIUS servers.
7. Add RADIUS to the authentication methods. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. This example will add RADIUS to the end of the list. See User authentication methods for information about adding methods to the beginning or middle of the list.

   ```
   (config)> add auth method end radius
   (config)>
   ```

8. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

LDAP

Your AnywhereUSB Plus device supports LDAP (Lightweight Directory Access Protocol), a protocol used for directory information services over an IP network. LDAP can be used with your AnywhereUSB Plus device for centralized authentication and authorization management for users who connect to the device. With LDAP support, the AnywhereUSB Plus device acts as an LDAP client, which sends user credentials and connection parameters to an LDAP server. The LDAP server then authenticates the LDAP client requests and sends back a response message to the device.

When you are using LDAP authentication, you can have both local users and LDAP users able to log in to the device. To use LDAP authentication, you must set up a LDAP server that is accessible by the AnywhereUSB Plus device prior to configuration. The process of setting up a LDAP server varies by the server environment.

This section contains the following topics:

LDAP user configuration

When configured to use LDAP support, the AnywhereUSB Plus device uses a remote LDAP server for user authentication (password verification) and authorization (assigning the access level of the user). Additional LDAP servers can be configured as backup servers for user authentication.

This section outlines how to configure a LDAP server to be used for user authentication on your AnywhereUSB Plus device.

There are several different implementations of LDAP, including Microsoft Active Directory. This section uses OpenLDAP as an example configuration. Other implementations of LDAP will have different configuration methods.

Example OpenLDAP configuration

With OpenLDAP, users can be configured in a text file using the LDAP Data Interchange Format (LDIF). In this case, we will be using a file called **add_user.ldif**.

1. Create the **add_user.ldif** file in a text editor. For example:

   ```
   $ gedit ./add_user.ldif
   ```

2. Add users to the file using the following format:

   ```
   dn: uid=john,dc=example,dc=com
   objectClass: inetOrgPerson
   cn: John Smith
   sn: Smith
   uid: john
   userPassword: password
   ou: admin serial
   ```

   - The value of **uid** and **userPassword** must correspond to the username and password used to log into the AnywhereUSB Plus device.
   - The **ou** attribute is optional. If used, the value must correspond to authentication groups configured on your AnywhereUSB Plus. Alternatively, if the user is also configured as a local user on the AnywhereUSB Plus device and the LDAP server authenticates the user but does not return any groups, the local configuration determines the list of groups. See Authentication groups for more information about authentication groups.

   Other attributes may be required by the user's objectClass. Any objectClass may be used as long it allows the **uid**, **userPassword**, and **ou** attributes.

3. Save and close the file.
4. Add the user to the OpenLDAP server:

   ```
   $ ldapadd -x -H 'ldap:///' -D 'cn=admin,dc=example,dc=com' -W -f add_
   user.ldif
   adding new entry "uid=john,dc=example,dc=com"
   ```

5. Verify that the user has been added by performing an LDAP search:

   ```
   $ ldapsearch -x -LLL -H 'ldap:///' -b 'dc=example,dc=com'
   uid=john
   dn: uid=john,dc=example,dc=com
   ```

```
objectClass: inetOrgPerson
cn: John Smith
sn: Smith
uid: john
ou: admin serial
```

LDAP server failover and fallback to local configuration

In addition to the primary LDAP server, you can also configure your AnywhereUSB Plus device to use backup LDAP servers. Backup LDAP servers are used for authentication requests when the primary LDAP server is unavailable.

Falling back to local authentication

With user authentication methods, you can configure your AnywhereUSB Plus device to use multiple types of authentication. For example, you can configure both LDAP authentication and local authentication, so that local authentication can be used as a fallback mechanism if the primary and backup LDAP servers are unavailable. Additionally, users who are configured locally but are not configured on the LDAP server are still able to log into the device. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned; therefore if you want to ensure that users are authenticated first through the LDAP server, and only authenticated locally if the LDAP server is unavailable or if the user is not defined on the LDAP server, then you should list the LDAP authentication method prior to the Local users authentication method.

See User authentication methods for more information about authentication methods.

If the LDAP servers are unavailable and the AnywhereUSB Plus device falls back to local authentication, only users defined locally on the device are able to log in. LDAP users cannot log in until the LDAP servers are brought back online.

Configure your AnywhereUSB Plus device to use an LDAP server

This section describes how to configure a AnywhereUSB Plus device to use an LDAP server for authentication and authorization.

Required configuration items

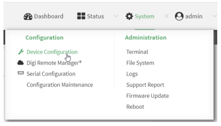
- Define the LDAP server IP address or domain name.
- Add LDAP as an authentication method for your AnywhereUSB Plus device.

Additional configuration items

- Whether other user authentication methods should be used in addition to the LDAP server, or if the LDAP server should be considered the authoritative login method.
- The LDAP server port. It is configured to 389 by default.
- Whether to use Transport Layer Security (TLS) when communicating with the LDAP server.
- The distinguished name (DN) and password used to communicate with the server.
- The distinguished name used to search to user base.
- The group attribute.
- The number of seconds to wait to receive a message from the server.
- Add additional LDAP servers in case the first LDAP server is unavailable.

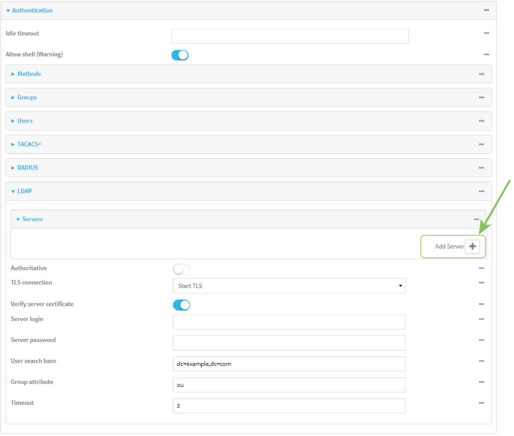
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Authentication** > **LDAP** > **Servers**.
4. Add LDAP servers:
   a. For **Add server**, click ✚.



Idle timeout

Allow shell (Warning) ●

Authoritative ○

TLS connection: Start TLS

Verify server certificate ●

Server login

Server password

User search base: dc=example,dc=com

Group attribute: ou

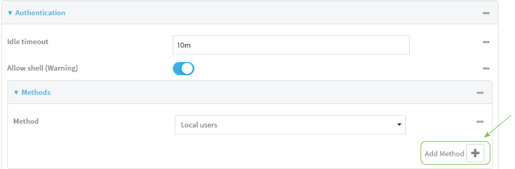
Timeout: 3

   b. For **Hostname**, type the hostname or IP address of the LDAP server.
   c. (Optional) Change the default **Port** setting to the appropriate port. Normally this should be left at the default setting of port 389.
   d. (Optional) Click ✚ again to add additional LDAP servers.
5. (Optional) Enable **Authoritative** to prevent other authentication methods from being used if LDAP authentication fails. Other authentication methods will only be used if the LDAP server is unavailable.
6. For **TLS connection**, select the type of TLS connection used by the server:
   - **Disable TLS**: Uses a non-secure TCP connection on the LDAP standard port, 389.
   - **Enable TLS**: Uses an SSL/TLS encrypted connection on port 636.
   - **Start TLS**: Makes a non-secure TCP connection to the LDAP server on port 389, then sends a request to upgrade the connection to a secure TLS connection. This is the preferred method for LDAP.
7. If **Enable TLS** or **Start TLS** are selected for **TLS connection**:
   - Leave **Verify server certificate** at the default setting of enabled to verify the server certificate with a known Certificate Authority.
   - Disable **Verify server certificate** if the server is using a self-signed certificate.

8. (Optional) For **Server login**, type a distinguished name (DN) that is used to bind to the LDAP server and search for users, for example **cn=user,dc=example,dc=com**. Leave this field blank if the server allows anonymous connections.
9. (Optional) For **Server password**, type the password used to log into the LDAP server. Leave this field blank if the server allows anonymous connections.
10. For **User search base**, type the distinguished name (DN) on the server to search for users. This can be the root of the directory tree (for example, **dc=example,dc=com**) or a sub-tree (for example. **ou=People,dc=example,dc=com**).
11. (Optional) For **Group attribute**, type the name of the user attribute that contains the list of AnywhereUSB Plus authentication groups that the authenticated user has access to. See LDAP user configuration for further information about the group attribute.
12. For **Timeout**, type or select the amount of time in seconds to wait for the LDAP server to respond. Allowed value is between **3** and **60** seconds.
13. Add LDAP to the authentication methods:

    a. Click **Authentication** > **Methods**.

    b. For **Add method**, click ✚.

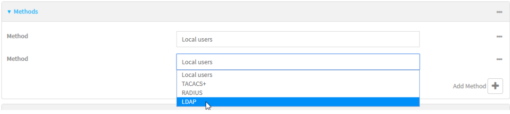


Idle timeout 10m

Allow shell (Warning)

Method Local users

    c. Select **LDAP** for the new method from the **Method** drop-down.

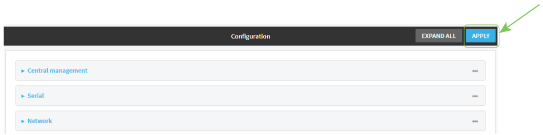


Method Local users

Method Local users

Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. See Rearrange the position of authentication methods for information about rearranging the position of the methods in the list.

14. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. (Optional) Prevent other authentication methods from being used if LDAP authentication fails. Other authentication methods will only be used if the LDAP server is unavailable.

   ```
   (config)> auth ldap authoritative true
   (config)>
   ```

4. Set the type of TLS connection used by the LDAP server:

   ```
   (config)> auth ldap tls value
   (config)>
   ```

   where *value* is one of:

   - **off**: Uses a non-secure TCP connection on the LDAP standard port, 389.
   - **on**: Uses an SSL/TLS encrypted connection on port 636.
   - **start_tls**: Makes a non-secure TCP connection to the LDAP server on port 389, then sends a request to upgrade the connection to a secure TLS connection. This is the preferred method for LDAP.

   The default is **off**.

5. If **tls** is set to **on** or **start_tls**, configure whether to verify the server certificate:

   ```
   (config)> auth ldap verify_server_cert value
   (config)>
   ```

   where *value* is either:

   - **true**: Verifies the server certificate with a known Certificate Authority.
   - **false**: Does not verify the certificate. Use this option if the server is using a self-signed certificate.

   The default is **true**.

6. Set the distinguished name (DN) that is used to bind to the LDAP server and search for users. Leave this option unset if the server allows anonymous connections.

   ```
   (config)> auth ldap bind_dn dn_value
   (config)>
   ```

   For example:

   ```
   (config)> auth ldap bind_dn cn=user,dc=example,dc=com
   (config)>
   ```

7. Set the password used to log into the LDAP server. Leave this option unset if the server allows anonymous connections.

   ```
   (config)> auth ldap bind_password password
   (config)>
   ```

8. Set the distinguished name (DN) on the server to search for users. This can be the root of the directory tree (for example, **dc=example,dc=com**) or a sub-tree (for example. **ou=People,dc=example,dc=com**).

```
(config)> auth ldap base_dn value
(config)>
```

9. (Optional) Set the name of the user attribute that contains the list of AnywhereUSB Plus authentication groups that the authenticated user has access to. See LDAP user configuration for further information about the group attribute.

   ```
   (config)> auth ldap group_attribute value
   (config)>
   ```

   For example:

   ```
   (config)> auth ldap group_attribute ou
   (config)>
   ```

10. Configure the amount of time in seconds to wait for the LDAP server to respond.

    ```
    (config)> auth ldap timeout value
    (config)>
    ```

    where *value* is any integer from **3** to **60**. The default value is **3**.

11. Add an LDAP server:

    a. Add the server:

    ```
    (config)> add auth ldap server end
    (config auth ldap server 0)>
    ```

    b. Enter the LDAP server's IP address or hostname:

    ```
    (config auth ldap server 0)> hostname hostname|ip-address
    (config auth ldap server 0)>
    ```

    c. (Optional) Change the default port setting to the appropriate port:

    ```
    (config auth ldap server 0)> port port
    (config auth ldap server 0)>
    ```

    d. (Optional) Repeat the above steps to add additional LDAP servers.

12. Add LDAP to the authentication methods. Authentication methods are attempted in the order they are listed until the first successful authentication result is returned. This example will add LDAP to the end of the list. See User authentication methods for information about adding methods to the beginning or middle of the list.

    ```
    (config)> add auth method end ldap
    (config)>
    ```

13. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

14. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

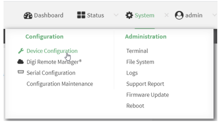
Disable shell access

To prohibit access to the shell prompt for all authentication groups, disable the **Allow shell** parameter.. This does not prevent access to the Admin CLI.

Note If shell access is disabled, re-enabling it will erase the device's configuration and perform a factory reset.

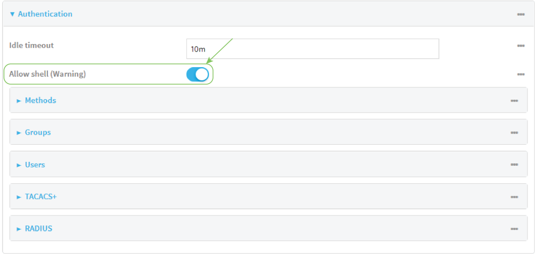
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.

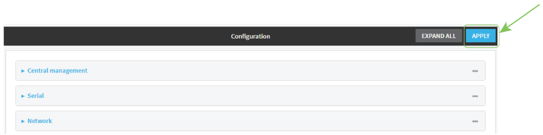
3. Click **Authentication**.
4. Click to disable **Allow shell**.


Idle timeout 10m

Allow shell (Warning)

   Note If shell access is disabled, re-enabling it will erase the device's configuration and perform a factory reset.

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Set the **allow_shell** parameter to **false**:

```
(config)> auth allow_shell false
```

   Note If shell access is disabled, re-enabling it will erase the device's configuration and perform a factory reset.

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

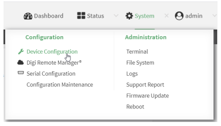
Set the idle timeout for AnywhereUSB Plus users

To configure the amount of time that the user's active session can be inactive before it is automatically disconnected, set the **Idle timeout** parameter.

By default, the Idle timeout is set to 10 minutes.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

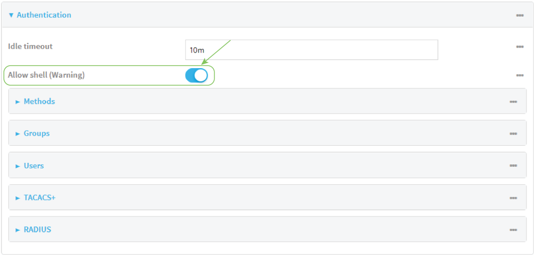


   The **Configuration** window is displayed.

3. Click **Authentication**.
4. For **Idle timeout**, enter the amount of time that the active session can be idle before the user is automatically logged out.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

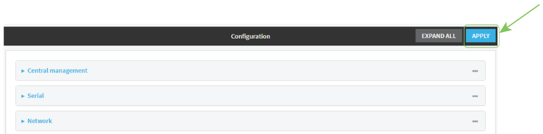
   For example, to set **Idle timeout** to ten minutes, enter **10m** or **600s**.



Idle timeout 10m

Allow shell (Warning)

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. At the config prompt, type:

   ```
   (config)# auth idle_timeout value
   ```

   where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number*{w|d|h|m|s}**.

   For example, to set **idle_timeout** to ten minutes, enter either **10m** or **600s**:

   ```
   (config)> auth idle_timeout  600s
   (config)>
   ```
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

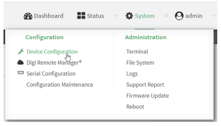
Example user configuration

Example 1: Administrator user with local authentication

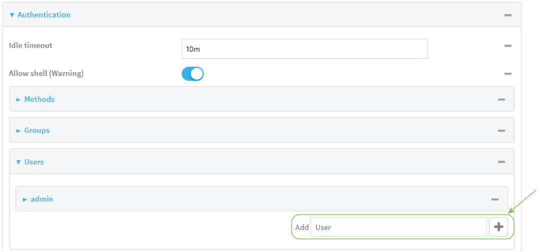
Goal: To create a user with administrator rights who is authenticated locally on the device.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

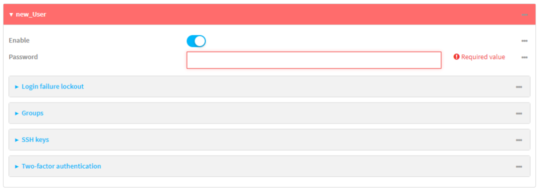
   

   The **Configuration** window is displayed.
3. Click **Authentication** > **Users**.
4. In **Add User:** enter a name for the user and click ✚.

   
   Idle timeout: 10m

   Allow shell (Warning): ●

   The user configuration window is displayed.

   
   Enable: ●

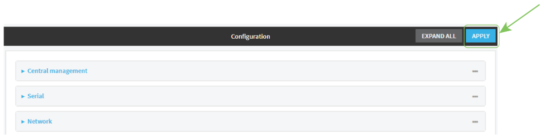
   Password: ❗ Required value

5. Enter a **Password** for the user.
6. Assign the user to the **admin** group:
   a. Click **Groups**.
   b. For **Add Group**, click ✚.
   c. For **Group**, select the **admin** group.
   d. Verify that the **admin** group has full administrator rights:
      i. Click **Authentication** > **Groups**.
      ii. Click **admin**.
      iii. Verify that the admin group has **Admin access** enabled. If not, click **Admin access** to enable.

        iv. Verify that **Access level** is set to **Full access**. If not, select **Full access**.

    e. Verify that **Local users** is one of the configured authentication methods:
        i. Click **Authentication** > **Methods**.
        ii. Verify that **Local users** is one of the methods listed in the list. If not:
            i. For **Add Method**, click ✚.
            ii. For **Method**, select **Local users**.

7. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Verify that the **admin** group has full administrator rights:

```
(config)> show auth group admin acl
admin
       enable true
       level full
...
(config)>
```

   If **admin** > **enable** is set to false:

```
(config)> auth group admin acl admin enable true
(config)>
```

   If **admin** > **level** is set to read-only:

```
(config)> auth group admin acl admin level full
(config)>
```

4. Verify that **local** is one of the configured authentication methods:

```
(config)> show auth method
0 local
(config)>
```

   If **local** is not listed:

```
(config)> add auth method end local
(config)>
```

5. Create the user. In this example, the user is being created with the username **adminuser**:

    ```
    (config)> add auth user adminuser
    (config auth user adminuser)>
    ```

6. Assign a password to the user:

    ```
    (config auth user adminuser)> password pwd
    (config auth user adminuser)>
    ```

7. Assign the user to the **admin** group:

    ```
    (config auth user adminuser)> add group end admin
    (config auth user adminuser)>
    ```

8. Save the configuration and apply the change:

    ```
    (config auth user adminuser)> save
    Configuration saved.
    >
    ```

9. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Example 2: RADIUS, TACACS+, and local authentication for one user

Goal: To create a user with administrator rights who is authenticated by using all three authentication methods.

In this example, when the user attempts to log in to the AnywhereUSB Plus device, user authentication will occur in the following order:

1. The user is authenticated by the RADIUS server. If the RADIUS server is unavailable,
2. The user is authenticated by the TACACS+ server. If both the RADIUS and TACACS+ servers are unavailable,
3. The user is authenticated by the AnywhereUSB Plus device using local authentication.

This example uses a FreeRadius 3.0 server running on ubuntu, and a TACACS+ server running on ubuntu. Server configuration may vary depending on the platforms or type of servers used in your environment.

WebUI

1. Configure a user on the RADIUS server:
   a. On the ubuntu machine hosting the FreeRadius server, open the **/etc/freeradius/3.0/users** file:

   ```
   $ sudo gedit /etc/freeradius/3.0/users
   ```

   b. Add a RADIUS user to the **users** file:

   ```
   admin1 Cleartext-Password := "password1"
          Unix-FTP-Group-Names := "admin"
   ```

   In this example:

   - The user's username is **admin1**.
   - The user's password is **password1**.
   - The authentication group on the AnywhereUSB Plus device, **admin**, is identified in the **Unix-FTP-Group-Names** parameter.

   c. Save and close the **users** file.
2. Configure a user on the TACACS+ server:
   a. On the ubuntu machine hosting the TACACS+ server, open the **/etc/tacacs+/tac_plus.conf** file:

   ```
   $ sudo gedit /etc/tacacs+/tac_plus.conf
   ```

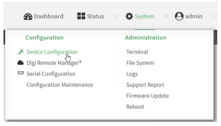
   b. Add a TACACS+ user to the **tac_plus.conf** file:

   ```
   user = admin1 {
          name ="Admin1 for TX64"
          pap = cleartext password1
          service = system {
                  groupname = admin
                  }
          }
   }
   ```

   In this example:

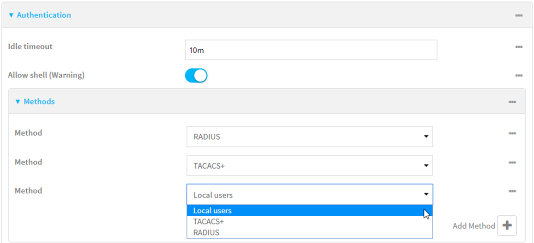
   - The user's username is **admin1**.
   - The user's password is **password1**.
   - The authentication group on the AnywhereUSB Plus device, **admin**, is identified in the **groupname** parameter.

   c. Save and close the **tac_plus.conf** file.
3. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
4. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



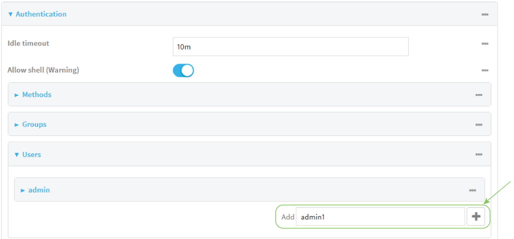
The **Configuration** window is displayed.

5. Configure the authentication methods:
   a. Click **Authentication** > **Methods**.
   b. For **Method**, select **RADIUS**.
   c. For **Add Method**, click ✚ to add a new method.
   d. For the new method, select **TACACS+**.
   e. Click ✚ to add another new method.
   f. For the new method, select **Local users**.



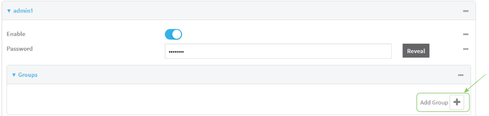
Idle timeout: 10m
Allow shell (Warning): on
Method: RADIUS
Method: TACACS+
Method: Local users

6. Create the local user:
   a. Click **Authentication** > **Users**.
   b. In **Add User:**, type **admin1** and click ✚.



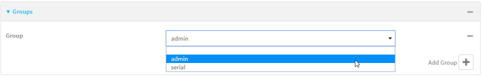
Idle timeout: 10m
Allow shell (Warning): on
Add: admin1

   c. For **password**, type **password1**.
   d. Assign the user to the **admin** group:
      i. Click **Groups**.
      ii. For **Add Group**, click ✚.



Enable: on
Password: ••••••••

      iii. For **Group**, select the **admin** group.

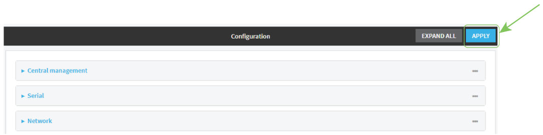


Group: admin

   a. Verify that the **admin** group has full administrator rights:
      i. Click **Authentication** > **Groups**.
      ii. Click **admin**.

   iii. Verify that the admin group has **Admin access** enabled. If not, click **Admin access** to enable.
   iv. Verify that **Access level** is set to **Full access**. If not, select **Full access**.

7. Click **Apply** to save the configuration and apply the change.



Command line

1. Configure a user on the RADIUS server:
   a. On the ubuntu machine hosting the FreeRadius server, open the **/etc/freeradius/3.0/users** file:

```
$ sudo gedit /etc/freeradius/3.0/users
```

   b. Add a RADIUS user to the **users** file:

```
admin1 Cleartext-Password := "password1"
       Unix-FTP-Group-Names := "admin"
```

   In this example:
   - The user's username is **admin1**.
   - The user's password is **password1**.
   - The authentication group on the AnywhereUSB Plus device, **admin**, is identified in the **Unix-FTP-Group-Names** parameter.

   c. Save and close the **users** file.

2. Configure a user on the TACACS+ server:
   a. On the ubuntu machine hosting the TACACS+ server, open the **/etc/tacacs+/tac_plus.conf** file:

```
$ sudo gedit /etc/tacacs+/tac_plus.conf
```

   b. Add a TACACS+ user to the **tac_plus.conf** file:

```
user = admin1 {
       name ="Admin1 for TX64"
       pap = cleartext password1
       service = system {
               groupname = admin
               }
       }
}
```

   In this example:
   - The user's username is **admin1**.
   - The user's password is **password1**.

- The authentication group on the AnywhereUSB Plus device, **admin**, is identified in the **groupname** parameter.

c. Save and close the **tac_plus.conf** file.

3. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

4. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

5. Configure the authentication methods:

   a. Determine the current authentication method configuration:

```
(config)> show auth method
0 local
(config)>
```

   This output indicates that on this example system, only local authentication is configured.

   b. Add RADIUS authentication to the beginning of the list:

```
(config)> add auth method 0 radius
(config)>
```

   c. Add TACACS+ authentication second place in the list:

```
(config)> add auth method 1 tacacs+(config)>
```

   d. Verify that authentication will occur in the correct order:

```
(config)> show auth method
0 radius
1 tacacs+
2 local
(config)>
```

6. Verify that the **admin** group has full administrator rights:

```
(config)> show auth group admin acl
admin
       enable true
       level full
...
(config)>
```

   If **admin** > **enable** is set to false:

```
(config)> auth group admin acl admin enable true
(config)>
```

If **admin** > **level** is set to read-only:

```
(config)> auth group admin acl admin level full
(config)>
```

7. Configure the local user:

   a. Create a local user with the username **admin1**:

   ```
   (config)> add auth user admin1
   (config auth user admin1)>
   ```

   b. Assign a password to the user:

   ```
   (config auth user adminuser)> password password1
   (config auth user adminuser)>
   ```

   c. Assign the user to the **admin** group:

   ```
   (config auth user adminuser)> add group end admin
   (config auth user adminuser)>
   ```

8. Save the configuration and apply the change:

   ```
   (config auth user adminuser)> save
   Configuration saved.
   >
   ```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Firewall

This chapter contains the following topics:

Firewall configuration

Firewall configuration includes the following configuration options:

- **Zones**: A zone is a firewall access group to which network interfaces can be added. You then use zones to configure packet filtering and access control lists for interfaces that are included in the zone. Preconfigured zones include:
    - **Any**: Matches any network interface, even if they are not assigned to this zone.
    - **Loopback**: Zone for interfaces that are used for communication between processes running on the device.
    - **Internal**: Used for interfaces connected to trusted networks. By default, the firewall will allow most access from this zone.
    - **External**: Used for interfaces to connect to untrusted zones, such as the internet. This zone has Network Address Translation (NAT) enabled by default. By default, the firewall will block most access from this zone.
    - **Edge**: Used for interfaces connected to trusted networks, where the device is a client on the edge of the network rather than a router or gateway.
    - **Setup**: Used for interfaces involved in the initial setup of the device. By default, the firewall will only allow this zone to access administration services.
    - **IPsec**: The default zone for IPsec tunnels.
    - **Dynamic routes**: Used for routes learned using routing services.
- **Port forwarding**: A list of rules that allow network connections to the AnywhereUSB Plus to be forwarded to other servers by translating the destination address.
- **Packet filtering**: A list of packet filtering rules that determine whether to accept or reject network connections that are forwarded through the AnywhereUSB Plus.
- **Custom rules**: A script that is run to install advanced firewall rules beyond the scope/capabilities of the standard device configuration.
- **Quality Of Service**: Quality of Service (QOS) options for bandwidth allocation and policy-based traffic shaping and prioritizing.

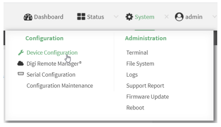
Create a custom firewall zone

In addition to the preconfigured zones, you can create your custom zones that can be used to configure packet filtering and access control lists for network interfaces.

To create a zone:

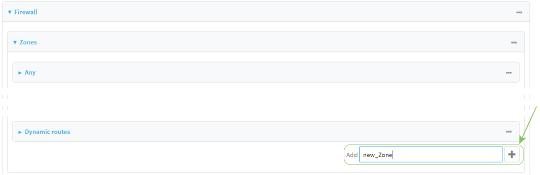
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

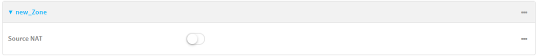


   The **Configuration** window is displayed.
3. Click **Firewall** > **Zones**.

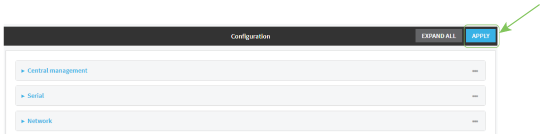
4. In **Add Zone**, enter a name for the zone and click ✚.



The firewall configuration window is displayed.



5. (Optional) If traffic on this zone will be forwarded from a private network to the internet, enable Network Address Translation (NAT).
6. Click **Apply** to save the configuration and apply the change.



See Configure the firewall zone for a network interface for information about how to configure network interfaces to use a zone.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add the new zone. For example, to add a zone named **my_zone**:

```
(config)> add firewall zone my_zone
(config firewall zone my_zone)>
```

4. (Optional) Enable Network Address Translation (NAT):

```
(config firewall zone my_zone)> src_nat true
(config firewall zone my_zone)>
```

5. Save the configuration and apply the change:

```
(config firewall zone my_zone)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

See Configure the firewall zone for a network interface for information about how to configure network interfaces to use a zone.

Configure the firewall zone for a network interface

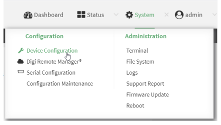
Firewall zones allow you to group network interfaces for the purpose of packet filtering and access control. There are several preconfigured firewall zones, and you can create custom zones as well. The firewall zone that a network interfaces uses is selected during interface configuration.

Delete a custom firewall zone

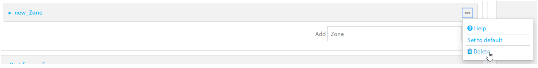
You cannot delete preconfigured firewall zones. To delete a custom firewall zone:

WebUI

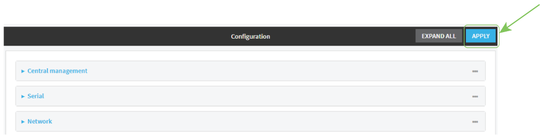
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Firewall** > **Zones**.
4. Click the menu icon (**...**) next to the appropriate custom firewall zone and select **Delete**.



5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Use the **del** command to delete a custom firewall rule. For example:

   ```
   (config)> del firewall zone my_zone
   ```
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Port forwarding rules

Most computers are protected by a firewall that prevents users on a public network from accessing servers on the private network. To allow a computer on the Internet to connect to a specific server on a private network, set up one or more port forwarding rules. Port forwarding rules provide mapping instructions that direct incoming traffic to the proper device on a LAN.

Configure port forwarding

Required configuration items

- The network interface for the rule.

  Network connections will only be forwarded if their destination address matches the IP address of the selected network interface.
- The public-facing port number that network connections must use for their traffic to be forwarded.
- The IP address of the server to which traffic should be forwarded.
- The port on the server to which traffic should be forwarded.

Additional configuration items

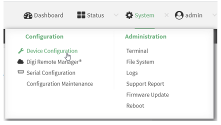
- A label for the port forwarding rule.
- The IP version (either IPv4 or IPv6) that incoming network connections must match.
- The protocols that incoming network connections must match.

- A white list of devices, based on either IP address or firewall zone, that are authorized to leverage this forwarding rule.

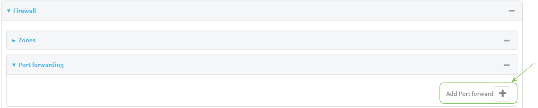
To configure a port forwarding rule:

WebUI

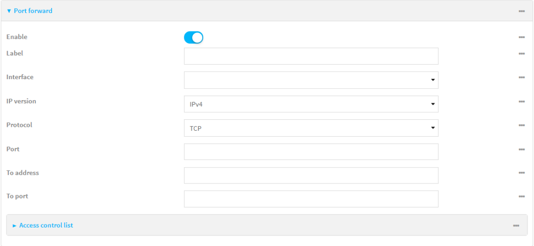
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Firewall** > **Port forwarding**.
4. For **Add port forward**, click ✚.

   The port forwarding rule configuration window is displayed.

   ▾ Port forward

   Enable ●

   Label

   Interface

   IP version IPv4

   Protocol TCP

   Port

   To address

   To port

   ▸ Access control list

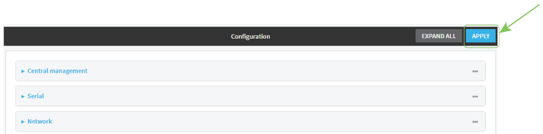
   Port forwarding rules are enabled by default. To disable, click to toggle off **Enable**.
5. (Optional) Type a **Label** that will be used to identify the rule.
6. For **Interface**, select the network interface for the rule.

   Network connections will only be forwarded if their destination address matches the IP address of the selected network interface.
7. For **IP version**, select either **IPv4** or **IPv6**.

   Network connections will only be forwarded if they match the selected IP version.
8. For **Protocol**, select the type of internet protocol.

   Network connections will only be forwarded if they match the selected protocol.
9. For **Port**, type the public-facing port number that network connections must use for their traffic to be forwarded.
10. For **To Address**, type the IP address of the server to which traffic should be forwarded.
11. For **To port**, type the port number of the port on the server to which traffic should be forwarded.

12. (Optional) Click **Access control list** to create a white list of devices that are authorized to leverage this forwarding rule, based on either the IP address or firewall zone:
    - To white list IP addresses:
        a. Click **Addresses**.
        b. For **Add Address**, enter an IP address and click ✚.
        c. Repeat for each additional IP address that should be white listed.
    - To specify firewall zones for white listing:
        a. Click **Zones**.
        b. For **Add zone**, click ✚.
        c. For **Zone**, select the appropriate zone.
        d. Repeat for each additional zone.
13. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. At the config prompt, type:

```
(config)> add firewall dnat end
(config firewall dnat 0)>
```

   Port forwarding rules are enabled by default. To disable the rule:

```
(config firewall dnat 0)> enable false
(config firewall dnat 0)>
```

4. Set the network interface for the rule.

```
(config firewall dnat 0)> interface
(config firewall dnat 0)>
```

   Network connections will only be forwarded if their destination address matches the IP address of this network interface.

a. Use the **?** to determine available interfaces:

```
(config firewall dnat 0)> interface ?

Interface: Network connections will only be forwarded if their
destination address matches the IP address of this network interface.
Format:
  defaultip
  defaultlinklocal
  eth1
  eth2
  loopback
Current value:

(config firewall dnat 0)> interface
```

b. Set the interface. For example:

```
(config firewall dnat 0)> interface eth1
(config firewall dnat 0)>
```

5. Set the IP version. Allowed values are **ipv4** and **ipv6**. The default is **ipv4**.

```
(config firewall dnat 0)> ip_version ipv6
(config firewall dnat 0)>
```

6. Set the public-facing port number that network connections must use for their traffic to be forwarded.

```
(config firewall dnat 0)> port port
(config firewall dnat 0)>
```

7. Set the type of internet protocol .

```
(config firewall dnat 0)> protocol value
(config firewall dnat 0)>
```

Network connections will only be forwarded if they match the selected protocol. Allowed values are **custom**, **tcp**, **tcpudp**, or **upd**. The default is **tcp**.

8. Set the IP address of the server to which traffic should be forwarded:
   - For IPv4 addresses:

   ```
   (config firewall dnat 0)> to_address ip-address
   (config firewall dnat 0)>
   ```

   - For IPv6 addresses:

   ```
   (config firewall dnat 0)> to_address6 ip-address
   (config firewall dnat 0)>
   ```

9. Set the public-facing port number that network connections must use for their traffic to be forwarded.

```
(config firewall dnat 0)> to_port port
(config firewall dnat 0)>
```

10. (Optional) To create a white list of devices that are authorized to leverage this forwarding rule, based on either the IP address or firewall zone, change to the acl node:

```
(config firewall dnat 0)> acl
(config firewall dnat 0 acl)>
```

- To white list an IP address:
  - For IPv4 addresses:

    ```
    (config firewall dnat 0 acl> add address end ip-address
    (config firewall dnat 0 acl)>
    ```

  - For IPv6 addresses:

    ```
    (config firewall dnat 0 acl> add address6 end ip-address
    (config firewall dnat 0 acl)>
    ```

  Repeat for each appropriate IP address.
- To specify the firewall zone for white listing:

  ```
  (config firewall dnat 0 acl)> add zone end zone
  ```

  Repeat for each appropriate zone.

  To view a list of available zones:

  ```
  (config firewall dnat 0 acl)> .. .. .. zone ?

  Zones: A list of groups of network interfaces that can be referred to
  by packet filtering rules
  and access control lists.

   Additional Configuration
   ----------------------------------------------------------------------
  ---------
    any
    dynamic_routes
    edge
    external
    internal
    ipsec
    loopback
    setup

  (config firewall dnat 0 acl)>
  ```

11. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

12. Type **exit** to exit the Admin CLI.

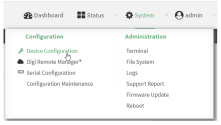
    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Delete a port forwarding rule

To delete a port forwarding rule:

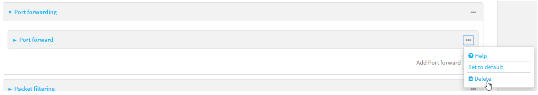
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

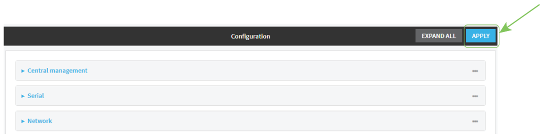


   The **Configuration** window is displayed.

3. Click **Firewall** > **Port forwarding**.
4. Click the menu icon (**...**) next to the appropriate port forwarding rule and select **Delete**.



5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Determine the index number of the port forwarding rule you want to delete:

```
(config)> show firewall dnat
0
        acl
                no address
                no zone
        enable true
        interface
        ip_version ipv4
        label IPv4 port forwarding rule
        port 10000
        protocol tcp
        to_address6 10.10.10.10
        to_port 10001

1
        acl
                no address6
                no zone
        enable false
        interface
        ip_version ipv6
        label IPv6 port forwarding rule
        port 10002
        protocol tcp
        to_address6 c097:4533:bd63:bb12:9a6f:5569:4b53:c29a
        to_port 10003
(config)>
```

4. To delete the rule, use the index number with the **del** command. For example:

```
(config)> del firewall dnat 1
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Packet filtering

By default, one preconfigured packet filtering rule, **Allow all outgoing traffic**, is enabled and monitors traffic going to and from the AnywhereUSB Plus device. The predefined settings are intended to block unauthorized inbound traffic while providing an unrestricted flow of outgoing data. You can modify the default packet filtering rule and create additional rules to define how the device accepts or rejects traffic that is forwarded through the device.

Configure packet filtering

Required configuration items

- The action that the packet filtering rule will perform, either **Accept**, **Reject**, or **Drop**.
- The source firewall zone: Packets originating from interfaces on this zone will be monitored by this rule.
- The destination firewall zone: Packets destined for interfaces on this zone will be accepted, rejected, or dropped by this rule.

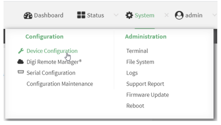
Additional configuration requirements

- A label for the rule.
- The IP version to be matched, either **IPv4**, **IPv6**, or **Any**.
- The protocol to be matched, one of:
    - **TCP**
    - **UDP**
    - **ICMP**
    - **ICMP6**
    - **Any**

To configure a packet filtering rule:

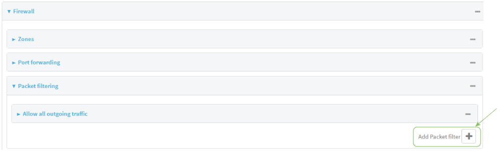
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

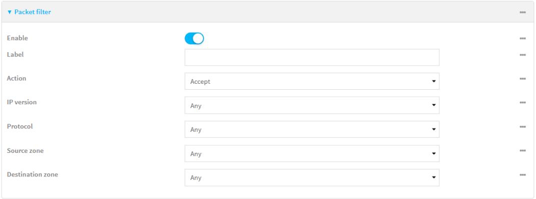
   The **Configuration** window is displayed.

3. Click **Firewall** > **Packet filtering**.
    - To create a new packet filtering rule, for **Add packet filter**, click ✚.

    - To edit the default packet filtering rule or another existing packet filtering rule, click to expand the rule.

   The packet filtering rule configuration window is displayed.

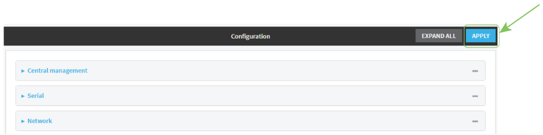
   

   Enable ●
   Label
   Action Accept
   IP version Any
   Protocol Any
   Source zone Any
   Destination zone Any

   Packet filters are enabled by default. To disable, click to toggle off **Enable**.
4. (Optional) Type a **Label** that will be used to identify the rule.
5. For **Action**, select one of:
    - **Accept**: Allows matching network connections.
    - **Reject**: Blocks matching network connections, and sends an ICMP error if appropriate.
    - **Drop**: Blocks matching network connections, and does not send a reply.
6. Select the **IP version**.
7. Select the **Protocol**.
8. For **Source zone**, select the firewall zone that will be monitored by this rule for incoming connections from network interfaces that are a member of this zone.

   See Firewall configuration for more information about firewall zones.
9. For **Destination zone**, select the firewall zone. Packets destined for network interfaces that are members of this zone will either be accepted, rejected or dropped by this rule.

   See Firewall configuration for more information about firewall zones.
10. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

To edit the default packet filtering rule or another existing packet filtering rule:

a. Determine the index number of the appropriate packet filtering rule:

```
(config)> show firewall filter
0
    action accept
    dst_zone any
    enable true
    ip_version any
    label Allow all outgoing traffic
    protocol any
    src_zone internal
1
    action drop
    dst_zone internal
    enable true
    ip_version any
    label myfilter
    protocol any
    src_zone external
(config)>
```

b. Select the appropriate rule by using its index number:

```
(config)> firewall filter 1
(config firewall filter 1)>
```

To create a new packet filtering rule:

```
(config)> add firewall filter end
(config firewall filter 1)>
```

Packet filtering rules are enabled by default. To disable the rule:

```
(config firewall filter 1)> enable false
(config firewall filter 1)>
```

3. (Optional) Set the label for the rule.

```
(config firewall filter 1)> label "My filter rule"
(config firewall filter 1)>
```

4. Set the action to be performed by the filter rule.

```
(config firewall filter 1)> action value
(config firewall filter 1)>
```

where *value* is one of:

- **accept**: Allows matching network connections.
- **reject**: Blocks matching network connections, and sends an ICMP error if appropriate.
- **drop**: Blocks matching network connections, and does not send a reply.

5. Set the firewall zone that will be monitored by this rule for incoming connections from network interfaces that are a member of this zone:

   See Firewall configuration for more information about firewall zones.

```
(config firewall filter 1)> src_zone my_zone
(config firewall filter 1)>
```

6. Set the destination firewall zone. Packets destined for network interfaces that are members of this zone will either be accepted, rejected or dropped by this rule.

   See Firewall configuration for more information about firewall zones.

```
(config firewall filter 1)> dst_zone my_zone
(config firewall filter 1)>
```

7. Set the IP version.

```
(config firewall filter 1)> ip_version value
(config firewall filter 1)>
```

where *value* is one of:

- **any**
- **ipv4**
- **ipv6**
- The default is **any**.

8. Set the protocol.

```
(config firewall filter 1)> protocol value
(config firewall filter 1)>
```

where value is one of:

- **any**
- **icmp**
- **icmpv6**
- **tcp**
- **upd**

The default is **any**.

9. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

10. Type **exit** to exit the Admin CLI.

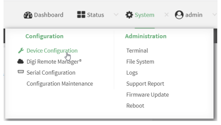
    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Enable or disable a packet filtering rule

To enable or disable a packet filtering rule:

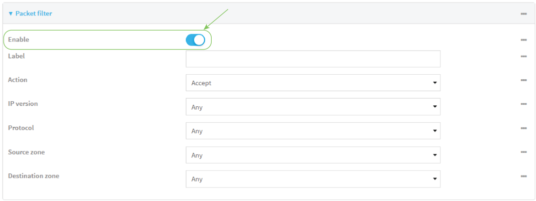
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



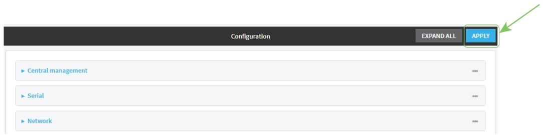
   The **Configuration** window is displayed.

3. Click **Firewall** > **Packet filtering**.
4. Click the appropriate packet filtering rule.
5. Click **Enable** to toggle the rule between enabled and disabled.



Enable
Label
Action: Accept
IP version: Any
Protocol: Any
Source zone: Any
Destination zone: Any

6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Determine the index number of the appropriate port forwarding rule:

```
(config)> show firewall filter
0
    action accept
    dst_zone any
    enable true
    ip_version any
    label Allow all outgoing traffic
    protocol any
    src_zone internal
1
    action drop
    dst_zone internal
    enable true
    ip_version any
    label My packet filter
    protocol any
    src_zone external
(config)>
```

4. To enable a packet filtering rule, use the index number with the **enable true** command. For example:

```
(config)> firewall filter 1 enable true
```

5. To disable a packet filtering rule, use the index number with the **enable false** command. For example:

```
(config)> firewall filter 1 enable false
```

6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

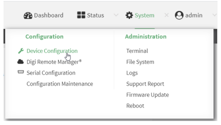
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Delete a packet filtering rule

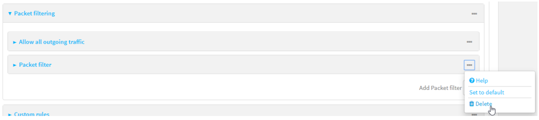
To delete a packet filtering rule:

☰ WebUI

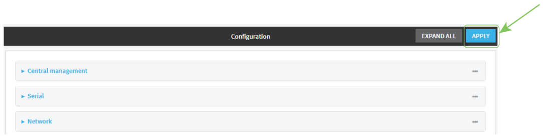
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Firewall** > **Packet filtering**.
4. Click the menu icon (**...**) next to the appropriate packet filtering rule and select **Delete**.

5. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Determine the index number of the packet filtering rule you want to delete:

```
(config)> show firewall filter
0
    action accept
    dst_zone any
    enable true
    ip_version any
    label Allow all outgoing traffic
    protocol any
    src_zone internal
1
    action drop
    dst_zone internal
    enable true
    ip_version any
    label My packet filter
    protocol any
    src_zone external
(config)>
```

4. To delete the rule, use the index number with the **del** command. For example:

```
(config)> del firewall filter 1
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

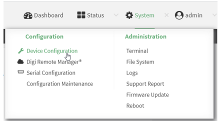
Configure custom firewall rules

Custom firewall rules consist of a script of shell commands that can be used to install firewall rules, ipsets, and other system configuration. These commands are run whenever system configuration changes occur that might cause changes to the firewall.

To configure custom firewall rules:

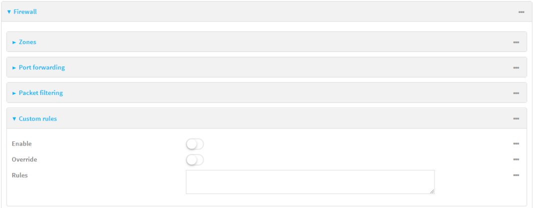
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



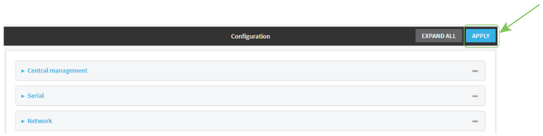
   The **Configuration** window is displayed.

3. Click **Firewall** > **Custom rules**.



4. **Enable** the custom rules.
5. (Optional) Enable **Override** to override all preconfigured firewall behavior and rely solely on the custom firewall rules.
6. For **Rules**, type the shell command that will execute the custom firewall rules script.

7. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable custom firewall rules:

```
(config)> firewall custom enable true
(config)>
```

4. (Optional) Instruct the device to override all preconfigured firewall behavior and rely solely on the custom firewall rules:

```
(config)> firewall custom override true
(config)>
```

5. Set the shell command that will execute the custom firewall rules script:

```
(config)> firewall custom rules "shell-command"
(config)>
```

6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure Quality of Service options

Quality of Service (QoS) options allow you to manage the traffic performance of various services, such as Voice over IP (VoIP), cloud computing, traffic shaping, traffic prioritizing, and bandwidth allocation. When configuring QOS, you can only control the queue for outgoing packets on each interface (egress packets), not what is received on the interface (packet ingress).

A QoS *binding* contains the policies and rules that apply to packets exiting the AnywhereUSB Plus device on the binding's interface. By default, the AnywhereUSB Plus device has two preconfigured QoS bindings, **Outbound** and **Inbound**. These bindings are an example configuration designed for a typical VoIP site:

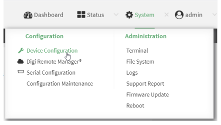
- **Outbound** provides an example of matching packets as they are routed from the device onto the WAN interface.
- **Inbound** provides an example of matching packets as they are routed from the device onto a LAN interface.

These example bindings are disabled by default.

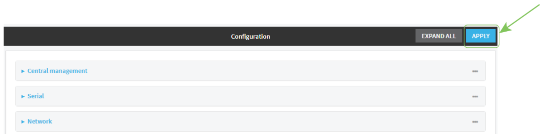
Enable the preconfigured bindings

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Firewall** > **Quality of Service**.
4. Click to expand either **Outbound** or **Inbound**.
5. **Enable** the binding.
6. Select an **Interface**.
7. Examine the remaining default settings and modify as appropriate for your network.
8. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Enable one of the preconfiged bindings:
   - To enable the Outbound binding:

     ```
     (config)> firewall qos 0 enable true
     (config)>
     ```
   - To enable the Inbound binding:

     ```
     (config)> firewall qos 1 enable true
     (config)>
     ```
4. Set the interface for the binding. Use the index number of the binding; for example, to set the interface for the Outbound binding:

a. Use the **?** to determine available interfaces:

```
(config)> firewall qos 0 interface ?

Interface: The network interface.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config)> firewall qos 0 interface
```

b. Set the interface. For example:

```
(config)> firewall qos 0 interface /network/interface/eth1
(config)>
```

5. Examine the remaining default settings and modify as appropriate for your network.
6. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

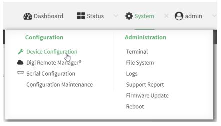
7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Create a new binding

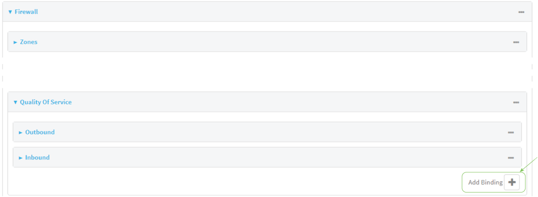
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

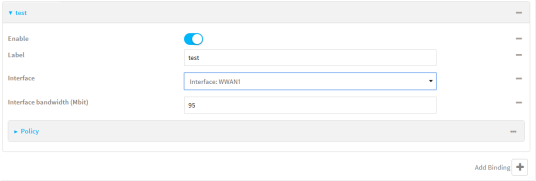


   The **Configuration** window is displayed.
3. Click **Firewall** > **Quality of Service**.

4. For **Add Binding**, click ✚.



The quality of service binding configuration window is displayed.



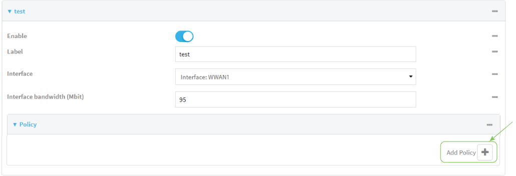
Enable: ●
Label: test
Interface: Interface: WWAN1
Interface bandwidth (Mbit): 95

5. **Enable** the binding.
6. (Optional) Type a **Label** for the binding.
7. Select an **Interface** to queue egress packets on. The binding will only match traffic that is being sent out on this interface.
8. (Optional) For **Interface bandwidth (Mbit)**, set the maximum egress bandwidth of the interface, in megabits, allocated to this binding. Typically, this should be 95% of the available bandwidth. Allowed value is any integer between **1** and **1000**.
9. Create a policy for the binding:

   At least one policy is required for each binding. Each policy can contain up to 30 rules.

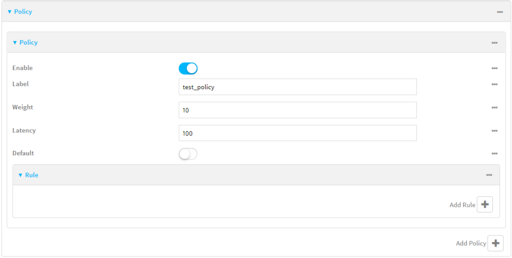
   a. Click to expand **Policy**.

   b. For **Add Policy**, click ✚.



Enable: ●
Label: test
Interface: Interface: WWAN1
Interface bandwidth (Mbit): 95

   The QoS binding policy configuration window is displayed.



Enable: ●
Label: test_policy
Weight: 10
Latency: 100
Default: ○

   New QoS binding policies are enabled by default. To disable, click **Enable**.

   c. (Optional) Type a **Label** for the binding policy.

d. For **Weight**, type a value for the amount of available bandwidth allocated to the policy, relative to other policies for this binding.

   The larger the weight, with respect to the other policy weights, the larger portion of the maximum bandwidth is available for this policy. For example, if a binding contains three policies, and each policy contains a weight of 10, each policy will be allocated one third of the total interface bandwidth.

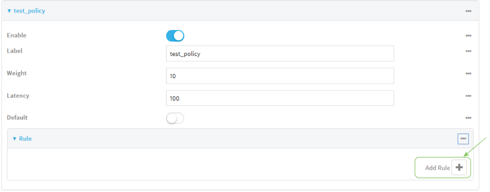
e. For **Latency**, type the maximum delay before the transmission of packets. A lower latency means that the packets will be scheduled more quickly for transmission.

f. Select **Default** to identify this policy as a fall-back policy. The fall-back policy will be used for traffic that is not matched by any other policy. If there is no default policy associated with this binding, packets that do not match any policy rules will be dropped.

g. If **Default** is disabled, you must configure at least one rule:

   i. Click to expand **Rule**.

   ii. For **Add Rule**, click ✚.



▼ test_policy

Enable ●

Label: test_policy

Weight: 10

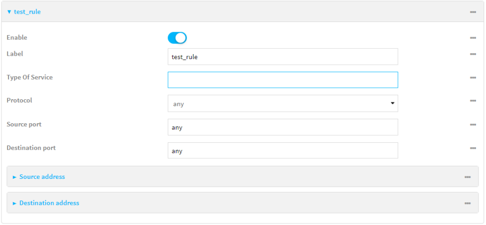
Latency: 100

Default ○

▼ Rule

Add Rule ✚

   The QoS binding policy rule configuration window is displayed.



▼ test_rule

Enable ●

Label: test_rule

Type Of Service:

Protocol: any

Source port: any

Destination port: any

► Source address

► Destination address

   New QoS binding policy rules are enabled by default. To disable, click **Enable**.

   iii. (Optional) Type a **Label** for the binding policy rule.

   iv. For **Type Of Service**, type the value of the Type of Service (ToS) packet header that defines packet priority. If unspecified, this field is ignored.

   See https://www.tucny.com/Home/dscp-tos for a list of common TOS values.

   v. For **Protocol**, select the IP protocol matching criteria for this rule.

   vi. For **Source port**, type the port, or **any**, as a source traffic matching criteria.

   vii. For **Destination port**, type the port, or **any**, as a destination traffic matching criteria.

   viii. Click to expand **Source address** and select the **Type**:

   - **Any**: Source traffic from any address will be matched.
   - **Interface**: Only traffic from the selected **Interface** will be matched.
   - **IPv4 address**: Only traffic from the IP address typed in **IPv4 address** will be matched. Use the format ***IPv4_address***[/***netmask***], or use **any** to match any IPv4 address.

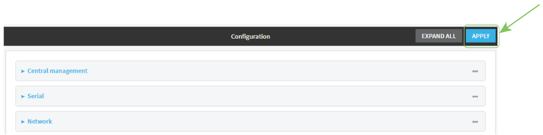
        - **IPv6 address**: Only traffic from the IP address typed in **IPv6 address** will be matched. Use the format ***IPv6_address*[/*prefix_length*]**, or use **any** to match any IPv6 address.
        - **MAC address**: Only traffic from the MAC address typed in **MAC address** will be matched.

    ix. Click to expand **Destination address** and select the **Type**:

        - **Any**: Traffic destined for anywhere will be matched.
        - **Interface**: Only traffic destined for the selected **Interface** will be matched.
        - **IPv4 address**: Only traffic destined for the IP address typed in **IPv4 address** will be matched. Use the format ***IPv4_address***[/***netmask*]**, or use **any** to match any IPv4 address.
        - **IPv6 address**: Only traffic destined for the IP address typed in **IPv6 address** will be matched. Use the format ***IPv6_address*[/*prefix_length*]**, or use **any** to match any IPv6 address.

    Repeat to add a new rule. Up to 30 rules can be configured.

10. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add a binding:

   ```
   (config)> add firewall qos end
   (config firewall qos 2)>
   ```

   New binding are enabled by default. To disable:

   ```
   (config firewall qos 2)> enable false
   (config firewall qos 2)>
   ```
4. (Optional) Set a label for the new binding:

   ```
   (config firewall qos 2)> label my_binding
   (config firewall qos 2)>
   ```
5. Set the interface to queue egress packets on. The binding will only match traffic that is being sent out on this interface:
   a. Use the **?** to determine available interfaces:

      ```
      (config firewall qos 2)> interface ?

      Interface: The network interface.
      Format:
        /network/interface/defaultip
        /network/interface/defaultlinklocal
        /network/interface/eth1
        /network/interface/eth2
        /network/interface/loopback
      Current value:

      (config firewall qos 2)> interface
      ```
   b. Set the interface. For example:

      ```
      (config firewall qos 2)> interface /network/interface/eth1
      (config firewall qos 2)>
      ```
6. (Optional) Set the maximum egress bandwidth of the interface, in megabits, allocated to this binding.

   ```
   (config firewall qos 2)> bandwidth int
   (config firewall qos 2)>
   ```

where *int* is an integer between **1** and **1000**. Typically, this should be 95% of the available bandwidth. The default is **95**.

7. Create a policy for the binding:

   At least one policy is required for each binding. Each policy can contain up to 30 rules.

   a. Change to the policy node of the configuration:

   ```
   (config firewall qos 2)> policy
   (config firewall qos 2 policy)>
   ```

   b. Add a policy:

   ```
   (config firewall qos 2 policy)> add end
   (config firewall qos 2 policy 0)>
   ```

   New QoS binding policies are enabled by default. To disable:

   ```
   (config firewall qos 2 policy 0)> enable false
   (config firewall qos 2 policy 0)>
   ```

   c. (Optional) Set a label for the new binding policy:

   ```
   (config firewall qos 2 policy 0)> label my_binding_policy
   (config firewall qos 2 policy 0)>
   ```

   d. Set a value for the amount of available bandwidth allocated to the policy, relative to other policies for this binding.

   The larger the weight, with respect to the other policy weights, the larger portion of the maximum bandwidth is available for this policy. For example, if a binding contains three policies, and each policy contains a weight of 10, each policy will be allocated one third of the total interface bandwidth.

   ```
   (config firewall qos 2 policy 0)> weight int
   (config firewall qos 2 policy 0)>
   ```

   where *int* is any integer between **1** and **65535**. The default is **10**.

   e. Set the maximum delay before the transmission of packets. A lower number means that the packets will be scheduled more quickly for transmission.

   ```
   (config firewall qos 2 policy 0)> latency int
   (config firewall qos 2 policy 0)>
   ```

   where *int* is any integer, **1** or greater. The default is **100**.

   f. To identify this policy as a fall-back policy:

   ```
   (config firewall qos 2 policy 0)> default true
   (config firewall qos 2 policy 0)>
   ```

   The fall-back policy will be used for traffic that is not matched by any other policy. If there is no default policy associated with this binding, packets that do not match any policy rules will be dropped. If the policy is not a fall-back policy, you must configure at least one rule:

i. Change to the rule node of the configuration:

```
(config firewall qos 2 policy 0)> rule
(config firewall qos 2 policy 0 rule)>
```

ii. Add a rule:

```
(config firewall qos 2 policy 0 rule)> add end
(config firewall qos 2 policy 0 rule 0)>
```

New QoS binding policy rules are enabled by default. To disable:

```
(config firewall qos 2 policy 0 rule 0)> enable false
(config firewall qos 2 policy 0 rule 0)>
```

iii. (Optional) Set a label for the new binding policy rule:

```
(config firewall qos 2 policy 0 rule 0)> label my_binding_policy_
rule
(config firewall qos 2 policy 0 rule 0)>
```

iv. Set the value of the Type of Service (ToS) packet header that defines packet priority. If unspecified, this field is ignored.

```
(config firewall qos 2 policy 0 rule 0)> tos value
(config firewall qos 2 policy 0 rule 0)>
```

where value is a hexadecimal number. See https://www.tucny.com/Home/dscp-tos for a list of common TOS values.

v. Set the IP protocol matching criteria for this rule:

```
(config firewall qos 2 policy 0 rule 0)> protocol value
(config firewall qos 2 policy 0 rule 0)>
```

where *value* is one of **tcp**, **udp**, or **any**.

vi. Set the source port to define a source traffic matching criteria:

```
(config firewall qos 2 policy 0 rule 0)> srcport value
(config firewall qos 2 policy 0 rule 0)>
```

where *value* is the IP port number, a range of port numbers using the format *IP_port-IP_port*, or **any**.

vii. Set the destination port to define a destination matching criteria:

```
(config firewall qos 2 policy 0 rule 0)> dstport value
(config firewall qos 2 policy 0 rule 0)>
```

where *value* is the IP port number, a range of port numbers using the format *IP_port-IP_port*, or **any**.

viii. Set the source address type:

```
(config network qos 2 policy 0 rule 0)> src type value
(config network qos 2 policy 0 rule 0)>
```

where *value* is one of:

- **any**: Source traffic from any address will be matched.

  See Firewall configuration for more information about firewall zones.
- **interface**: Only traffic from the selected interface will be matched. Set the interface:

  i. Use the **?** to determine available interfaces:

  ```
  (config network qos 2 policy 0 rule 0)> src interface ?

  Interface: Match the IP address with the specified
  interface's network address.
  Format:
    /network/interface/defaultip
    /network/interface/defaultlinklocal
    /network/interface/eth1
    /network/interface/eth2
    /network/interface/loopback
  Current value:

  (config network qos 2 policy 0 rule 0)> src interface
  ```

  ii. Set the interface. For example:

  ```
  (config network qos 2 policy 0 rule 0)> src interface
  /network/interface/eth1
  (config network qos 2 policy 0 rule 0)>
  ```
- **address**: Only traffic from the IP address typed in **IPv4 address** will be matched. Set the address that will be matched:

  ```
  (config network qos 2 policy 0 rule 0)> src address value
  (config network qos 2 policy 0 rule 0)>
  ```

  where value uses the format ***IPv4_address***[/***netmask***], or **any** to match any IPv4 address.
- **address6**: Only traffic from the IP address typed in **IPv6 address** will be matched. Set the address that will be matched:

  ```
  (config network qos 2 policy 0 rule 0)> src address6 value
  (config network qos 2 policy 0 rule 0)>
  ```

  where value uses the format ***IPv6_address***[/***prefix_length***], or **any** to match any IPv6 address.
- **mac**: Only traffic from the MAC address typed in **MAC address** will be matched. Set the MAC address to be matched:

```
(config network qos 2 policy 0 rule 0)> src mac MAC_address
(config network qos 2 policy 0 rule 0)>
```

ix. Set the destination address type:

```
(config network qos 2 policy 0 rule 0)> dst type value
(config network qos 2 policy 0 rule 0)>
```

where *value* is one of:

- **any**: Traffic destined for anywhere will be matched.

  See Firewall configuration for more information about firewall zones.

- **interface**: Only traffic destined for the selected **Interface** will be matched. Set the interface:

  i. Use the **?** to determine available interfaces:

  ```
  (config network qos 2 policy 0 rule 0)> dst interface ?

  Interface: Match the IP address with the specified
  interface's network address.
  Format:
    /network/interface/defaultip
    /network/interface/defaultlinklocal
    /network/interface/eth1
    /network/interface/eth2
    /network/interface/loopback
  Current value:

  (config network qos 2 policy 0 rule 0)> dst interface
  ```

  ii. Set the interface. For example:

  ```
  (config network qos 2 policy 0 rule 0)> dst interface
  /network/interface/eth1
  (config network qos 2 policy 0 rule 0)>
  ```

- **address**: Only traffic destined for the IP address typed in **IPv4 address** will be matched. Set the address that will be matched:

  ```
  (config network qos 2 policy 0 rule 0)> src address value
  (config network qos 2 policy 0 rule 0)>
  ```

  where value uses the format ***IPv4_address***[/***netmask***], or **any** to match any IPv4 address.

- **address6**: Only traffic destined for the IP address typed in **IPv6 address** will be matched. Set the address that will be matched:

  ```
  (config network qos 2 policy 0 rule 0)> src address6 value
  (config network qos 2 policy 0 rule 0)>
  ```

where value uses the format ***IPv6_address***[**/*prefix_length***], or **any** to match any IPv6 address.

Repeat to add a new rule. Up to 30 rules can be configured.

8. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

System administration

This chapter contains the following topics:

Review device status

You can review the system of your device from either the **Status** page of the Web interface, or from the command line:

WebUI

To display system information:

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **Status**.

   A secondary menu appears, along with a status panel.
3. On the secondary menu, click to display the details panel for the status you want to view.

Command line

To display system information, use the show system command.

- Show basic system information:
  1. Log into the AnywhereUSB Plus command line as a user with Admin access.

     Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
  2. Enter **show system** at the prompt:

```
> show system

  Model                        : Digi AnywhereUSB Plus
  Serial Number                : AnywhereUSB Plus-000065
  SKU                          : AnywhereUSB Plus
  Hostname                     : AnywhereUSB Plus
  MAC                          : DF:DD:E2:AE:21:18

  Hardware Version             : 50001947-01 1P
  Firmware Version             : 21.2.39.67
  Alt. Firmware Version        : 21.2.39.67
  Bootloader Version           : 19.7.23.0-15f936e0ed

  Current Time                 : Fri, 26 Feb 2021 8:04:23 +0000
  CPU                          : 1.4%
  Uptime                       : 6 days, 6 hours, 21 minutes, 57 seconds
(541317s)
  Temperature                  : 40C

>
```

- Show more detailed system information:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Enter **show system verbose** at the prompt:

```
> show system verbose

  Model                    : Digi AnywhereUSB Plus
  Serial Number            : AnywhereUSB Plus-000065
  SKU                      : AnywhereUSB Plus
  Hostname                 : AnywhereUSB Plus
  MAC                      : DF:DD:E2:AE:21:18

  Hardware Version         : 50001947-01 1P
  Firmware Version         : 21.2.39.67
  Alt. Firmware Version    : 21.2.39.67
  Bootloader Version       : 19.7.23.0-15f936e0ed

  Schema Version           : 715
  Timezone                 : UTC
  Current Time             : Fri, 26 Feb 2021 8:04:23 +0000
  CPU                      : 1.4%
  Uptime                   : 6 days, 6 hours, 21 minutes, 57 seconds
(541317s)
  Temperature              : 40C

  Disk
  ----
  Load Average             : 0.09, 0.10, 0.08
  RAM Usage                : 127.843MB/1880.421MB(6%)
  Disk /etc/config Usage   : 18.421MB/4546.371MB(0%)
  Disk /opt Usage          : -4523.-46MB/549.304MB(-822%)
  Disk /overlay Usage      : MB/MB(%)
  Disk /tmp Usage          : 0.007MB/256.0MB(0%)
  Disk /var Usage          : 1.765MB/256.0MB(1%)

>
```

Configure system information

You can configure information related to your AnywhereUSB Plus device, such as providing a name and location for the device.

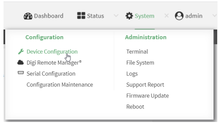
Configuration items

- A name for the device.
- The name of a contact for the device.
- The location of the device.
- A description of the device.
- A banner that will be displayed when users access terminal services on the device.

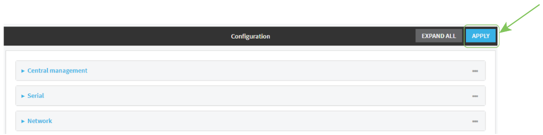
To enter system information:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **System**.
4. For **Name**, type a name for the device. This name will appear in log messages and at the command prompt.
5. For **Contact**, type the name of a contact for the device.
6. For **Location**, type the location of the device.
7. For **Banner**, type a banner message that will be displayed when users log into terminal services on the device.
8. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Set a name for the device. This name will appear in log messages and at the command prompt.

```
(config)> system name 192.168.3.1
192.168.3.1(config)>
```

4. Set the contact for the device:

```
192.168.3.1(config)> system contact "Jane User"
192.168.3.1(config)>
```

5. Set the location for the device:

```
192.168.3.1(config)> system location "9350 Excelsior Blvd., Suite 700,
Hopkins, MN"
192.168.3.1(config)>
```

6. Set the banner for the device. This is displayed when users access terminal services on the device.

```
192.168.3.1(config)> system banner "Welcome to the Digi AnywhereUSB Plus."
192.168.3.1(config)>
```

7. Save the configuration and apply the change:

```
192.168.3.1(config)> save
Configuration saved.
192.168.3.1>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Update system firmware

The AnywhereUSB Plus operating system firmware images consist of a single file with the following naming convention:

***platform-version*.bin**

For example, **AnywhereUSB Plus-21.2.39.67.bin**.

Manage firmware updates using Digi Remote Manager

If you have a network of many devices, you can use Digi Remote Manager **Profiles** to manage firmware updates. Profiles ensure all your devices are running the correct firmware version and that all newly installed devices are updated to that same version. For more information, see the **Profiles** section of the *Digi Remote Manager User Guide*.

Certificate management for firmware images

The system firmware files are signed to ensure that only Digi-approved firmware load onto the device. The AnywhereUSB Plus device validates the system firmware image as part of the update process and only successfully updates if the system firmware image can be authenticated.

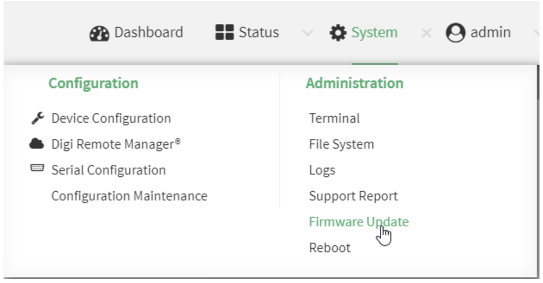
Downgrading

Downgrading to an earlier release of the firmware may result in the device configuration being erased.

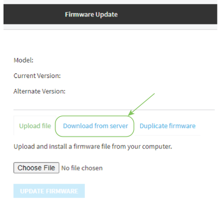
WebUI

Install firmware from the Digi firmware server

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Administration**, click **Firmware Update**.



3. Click **Download from server**.

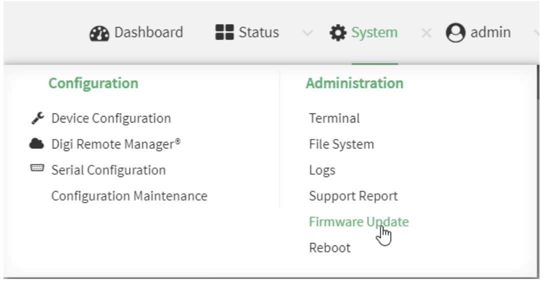


Upload and install a firmware file from your computer.

4. For **Version:**, select the appropriate version of the device firmware.
5. Click **Update Firmware**.

Update firmware from a local file

1. Download the AnywhereUSB Plus operating system firmware from the Digi Support FTP site to your local machine.
2. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
3. On the main menu, click **System**. Under **Administration**, click **Firmware Update**.



4. Click **Choose file**.
5. Browse to the location of the firmware on your local file system and select the file.
6. Click **Update Firmware**.

⌨ Command line

1. Download the AnywhereUSB Plus operating system firmware from the Digi Support FTP site to your local machine.
2. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
3. Load the firmware image onto the device:

   ```
   > scp host hostname-or-ip user username remote remote-path local local-path
   to local
   ```

   where:

   - *hostname-or-ip* is the hostname or ip address of the remote host.
   - *username* is the name of the user on the remote host.
   - *remote-path* is the path and filename of the file on the remote host that will be copied to the AnywhereUSB Plus device.
   - *local-path* is the location on the AnywhereUSB Plus device where the copied file will be placed.
4. Verify that the firmware file has been successfully uploaded to the device:

   ```
   > ls /etc/config/scripts
   -rw-r--r--    1 root     root      37511229 May 16 20:10 AnywhereUSB Plus-
   21.2.39.67.bin
   -rw-r--r--    1 root     root          2580 May 16 16:44 accns.json
   ...
   >
   ```
5. Update the firmware by entering the update firmware command, specifying the firmware file name.
6. Reboot the device to run the new firmware image using the reboot command.

   ```
   > reboot
   Rebooting system
   >
   ```
7. Once the device has rebooted, log into the AnywhereUSB Plus's command line as a user with Admin access and verify the running firmware version by entering the show system command.

   ```
   > show system

    Hostname                 : AnywhereUSB Plus
    FW Version               : 21.2.39.67
    MAC                      : 0040FF800120
    Model                    : Digi AnywhereUSB Plus
    Current Time             : Fri, 26 Feb 2021 8:04:23 +0000
    Uptime                   : 42 seconds (42s)

   >
   ```

Dual boot behavior

By default, the AnywhereUSB Plus device stores two copies of firmware in two flash memory banks:

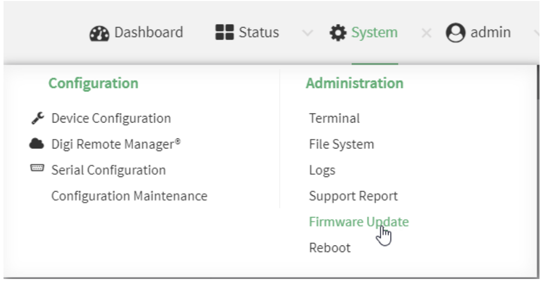
- The current firmware version that is used to boot the device.
- A copy of the firmware that was in use prior to your most recent firmware update.

When the device reboots, it will attempt to use the current firmware version. If the current firmware version fails to load after three consecutive attempts, it is marked as invalid and the device will use the previous firmware version stored in the alternate memory bank.

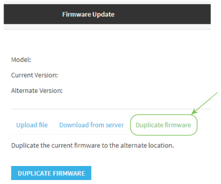
If the device consistently looses power during the boot process, this may result in the current firmware being marked as invalid and the device downgrading to a previous version of the firmware. As a result of this behavior, you can use the following procedure to guarantee that the same firmware is stored in both memory banks:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Administration**, click **Firmware Update**.



3. Click **Duplicate firmware**.



Duplicate the current firmware to the alternate location.

4. Click **Duplicate Firmware**.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Duplicate the firmware:

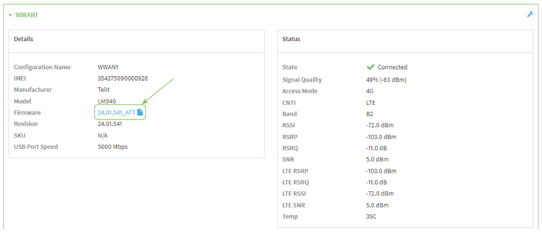
```
> system duplicate-firmware
>
```

Update cellular module firmware

You can update modem firmware by downloading firmware from the Digi firmware repository, or by uploading firmware from your local storage onto the device. You can also schedule modem firmware updates. See Schedule system maintenance tasks for details.

WebUI

1. (Optional) Download the appropriate modem firmware from the Digi repository to your local machine.
2. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
3. From the main menu, click **Status** > **Modems**.
4. Click the modem firmware version.



WWAN1

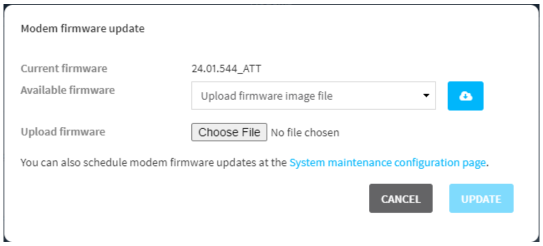
Details

Configuration Name	WWAN1
IMEI	354375090000926
Manufacturer	Telit
Model	LM940
Firmware	24.01.541_ATT
Revision	24.01.541
SKU	N/A
USB Port Speed	5000 Mbps

Status

State	✓ Connected
Signal Quality	49% (-83 dBm)
Access Mode	4G
CNTI	LTE
Band	B2
RSSI	-72.0 dBm
RSRP	-103.0 dBm
RSRQ	-11.0 dB
SNR	5.0 dBm
LTE RSRP	-103.0 dBm
LTE RSRQ	-11.0 dB
LTE RSSI	-72.0 dBm
LTE SNR	5.0 dBm
Temp	35C

The **Modem firmware update** window opens.



Modem firmware update

Current firmware 24.01.544_ATT

Available firmware Upload firmware image file

Upload firmware Choose File No file chosen

You can also schedule modem firmware updates at the System maintenance configuration page.

5. To update using firmware from the Digi firmware repository:
   a. Click to view available versions.
   b. For Available firmware, select the firmware.
6. To update using firmware from your local file system:
   a. Click **Choose File**.
   b. Select the firmware.
7. To schedule firmware updates, click **System maintenance configuration page**. See Schedule system maintenance tasks for details.
8. Click **Update**.

Command line

Update modem firmware over the air (OTA)

You can update your modem firmware by querying the Digi firmware repository to determine if there is new firmware available for your modem and performing an OTA modem firmware update:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Use the **modem firmware ota check** command to determine if new modem firmware is available on the Digi firmware repository.

```
> modem firmware ota check

Checking for latest ATT firmware ...
Retrieving modem firmware list ...
Newest firmware version available to download is '24.01.5x4_ATT'
Modem firmware update from '24.01.544_ATT' to '24.01.5x4_ATT' is needed
24.01.5x4_ATT
24.01.544_ATT

>
```

3. Use the **modem firmware ota list** command to list available firmware on the Digi firmware repository.

```
>  modem firmware ota list

Retrieving modem firmware list ...
25.20.664_CUST_044_3
25.20.666_CUST_067_1
25.20.663_CUST_040

>
```

4. Perform an OTA firmware update:

   - To perform an OTA firmware update by using the most recent available modem firmware from the Digi firmware repository, type:

   ```
   >  modem firmware ota update

   Checking for latest Generic firmware ...
   Retrieving modem firmware list ...
   Newest firmware version available to download is '25.20.666_CUST_067_
   1'
   Retrieving download location for modem firmware '25.20.666_CUST_067_1'
   ...

   >
   ```

   - To perform an OTA firmware update by using a specific version from the Digi firmware repository, use the **version** parameter to identify the appropriate firmware version as determined using the **modem firmware ota check** or **modem firmware ota list** command. For example::

```
>  modem firmware ota update version 24.01.5x4_ATT

Retrieving download location for modem firmware '24.01.5x4_ATT' ...
Downloading modem firmware '24.01.5x4_ATT' to '/opt/LE910C4_NF/Custom_
Firmware' ...
Modem firmware '24.01.5x4_ATT' downloaded
Updating modem firmware ...
Programming modem firmware ...

Found modem ...
Validate modem firmware ...
Getting ready for update ...
Stopping services ...
Running update pass 1 of 3 ...
Restarting services ...
-----------------------------
Successfully updated firmware
Modem firmware update complete

>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Update modem firmware by using a local firmware file

You can update your modem firmware by uploading a modem firmware file to your AnywhereUSB Plus device. Firmware should be uploaded to /opt/*MODEM_MODEL*/Custom_Firmware, for example, /opt/LM940/Custom_Firmware. Modem firmware can be downloaded from Digi at https://ftp1.digi.com/support/firmware/dal/carrier_firmware/. See Use the scp command for information about uploading files to the AnywhereUSB Plus device.

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Use the **modem firmware check** command to determine if new modem firmware is available on local device.

```
> modem firmware check

Checking for latest ATT firmware in flash ...
Newest firmware version available in flash is '05.05.58.00_ATT_005.026_000'
Modem firmware up to date
05.05.58.00_ATT_005.026_000

> modem firmware check
```

3. Use the **modem firmware list** command to list available firmware on the AnywhereUSB Plus device.

```
>  modem firmware list

ATT, 24.01.544_ATT, current
Generic, 24.01.514_Generic, image
Verizon, 24.01.524_Verizon, image
ATT, 24.01.544_ATT, image
Sprint, 24.01.531-B003_Sprint, image

>
```

4. To perform an firmware update by using a local file, use the **version** parameter to identify the appropriate firmware version as determined using the **modem firmware check** or **modem firmware list** command. For example::

```
>  modem firmware update version 24.01.5x4_ATT

Updating modem firmware ...

-----------------------------
Successfully updated firmware
Modem firmware update complete

>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Reboot your AnywhereUSB Plus device

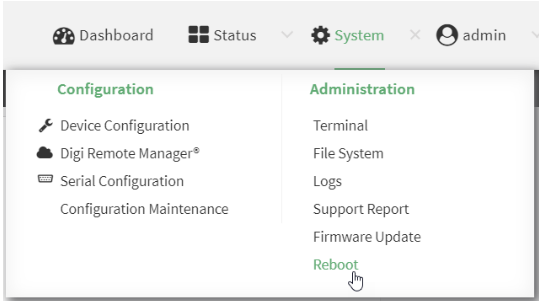
You can reboot the AnywhereUSB Plus device immediately or schedule a reboot for a specific time every day.

Note You may want to save your configuration settings to a file before rebooting. See Save configuration to a file.

Reboot your device immediately

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. From the main menu, click **System**.
3. Click **Reboot**.



4. Click **Reboot** to confirm that you want to reboot the device.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

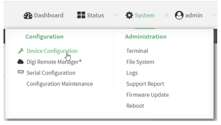
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the prompt, type:

```
> reboot
```

Schedule reboots of your device

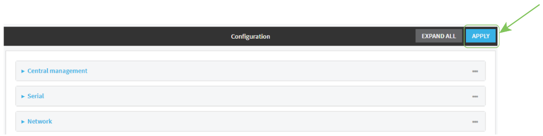
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Select **System** > **Scheduled tasks**.
4. For **Reboot time**, enter the time of the day that the device should reboot, using the format *HH*:*MM*. The device will reboot at this time every day.

   If a value is set for **Reboot time** but the device is unable to synchronize its time with an NTP server, the device will reboot after it has been up for 24 hours. See System time for information about configuring NTP servers.
5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Set the reboot time:

   ```
   (config>> system schedule reboot_time time
   (config)>
   ```

   where *time* is the time of the day that the device should reboot, using the format *HH*:*MM*. For example, the set the device to reboot at two in the morning every day:

   ```
   (config>> system schedule reboot_time 02:00
   (config)>
   ```

   If a value is set for **reboot_time** but the device is unable to synchronize its time with an NTP server, the device will reboot after it has been up for 24 hours. See System time for information about configuring NTP servers.
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Erase device configuration and reset to factory defaults

You can erase the device configuration in the WebUI, at the command line, or by using the **RESET** button on the device. Erasing the device configuration performs the following actions:

- Clears all configuration settings. When the device restarts, it uses the factory default configuration.
- Deletes all user files including Python scripts.
- Clears event and system log files.

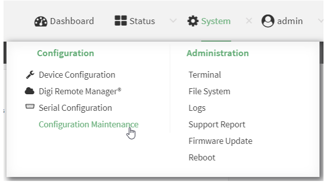
Additionally, if the **RESET** button is used to erase the configuration, pressing the **RESET** button a second time immediately after the device has rebooted:

- Erases all automatically generated certificates and keys.

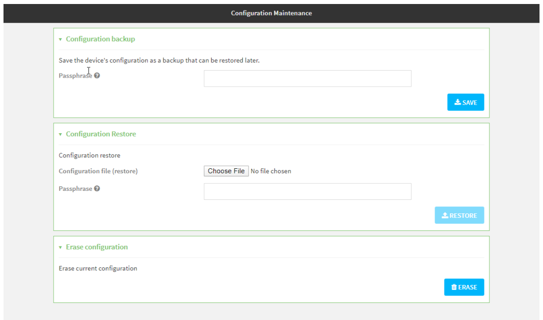
You can also reset the device to the default configuration without removing scripts, keys, and logfiles by using the **revert** command.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Configuration**, click **Configuration Maintenance**.



The **Configuration Maintenance** windows is displayed.



Save the device's configuration as a backup that can be restored later.

Passphrase

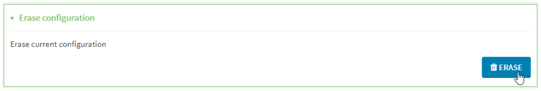
Configuration restore

Configuration file (restore) No file chosen

Passphrase

Erase current configuration

3. In the **Erase configuration** section, click **ERASE**.



Erase current configuration

4. Click **CONFIRM**.
5. After resetting the device:
   a. Connect to the AnywhereUSB Plus by using the serial port or by using an Ethernet cable to connect the AnywhereUSB Plus **ETH2** port to your PC.
   b. Log into the AnywhereUSB Plus:

      User name: Use the default user name: **admin**.

      Password: Use the unique password printed on the bottom label of the device (or the printed label included in the package).

      When you first log into the WebUI or the command line, you must change the password for the **admin** user. See Change the default password for the admin user for instructions.
   c. Reset the default password for the admin account. See Change the default password for the admin user for further information.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Enter the following:

   ```
   > system factory-erase
   ```
3. After resetting the device:
   a. Connect to the AnywhereUSB Plus by using the serial port or by using an Ethernet cable to connect the AnywhereUSB Plus **ETH2** port to your PC.
   b. Log into the AnywhereUSB Plus:

      User name: Use the default user name: **admin**.

      Password: Use the unique password printed on the bottom label of the device (or the printed label included in the package).

      When you first log into the WebUI or the command line, you must change the password for the **admin** user. See Change the default password for the admin user for instructions.
   c. Reset the default password for the admin account. See Change the default password for the admin user for further information.

Reset the device by using the Reset button

Note Using the reset button is the most extreme factory reset option.

If the AnywhereUSB Hub is physically accessible, you can use the **RESET** button on the Hub to restore the configuration to factory defaults. The restore process clears all current settings (including all previously stored client IDs and certificates), deletes all Hub and **AnywhereUSB Manager** keys, resets the password for the administrative user, and restores the settings to the factory defaults.

After you restore the factory defaults on a Hub, none of the existing **AnywhereUSB Managers** will be able to connect to the Hub. When the Hub is restored, the Hub creates a new Hub certificate, which will not be accepted by the existing **AnywhereUSB Managers**. To ensure that the new Hub certificate is accepted by the existing **AnywhereUSB Managers**, you must re-deploy the Hub.

1. Locate the **RESET** button on your device.
   - AnywhereUSB Plus 2 port: The **RESET** button is on the front panel.
   - AnywhereUSB Plus 8 port: The **RESET** button is on the back panel.
   - AnywhereUSB Plus 24 port: The **RESET** button is on the side of the device.
2. Press and hold the **RESET** button for about 10 seconds, until all the USB LEDs blink twice.
3. Release the **RESET** button. The Hub automatically reboots.
4. After resetting the device, you must re-deploy the Hub.

Reset the device with the revert command

You can reset the device to the default configuration without removing scripts, keys, and logfiles by using the **revert** command:

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. At the config prompt, enter **revert**:

```
(config)> revert
(config)>
```

4. Set the password for the admin user prior to saving the changes:

```
(config)> auth user admin password pwd
(config)>
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the AnywhereUSB Plus device to use custom factory default settings

You can configure your AnywhereUSB Plus device to use custom factory default settings. This way, when you erase the device's configuration, the device will reset to your custom configuration rather than to the original factory defaults.

Required configuration items

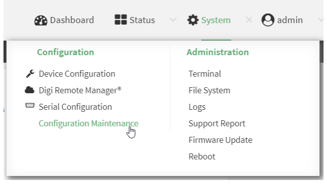
- Custom factory default file

☰ WebUI

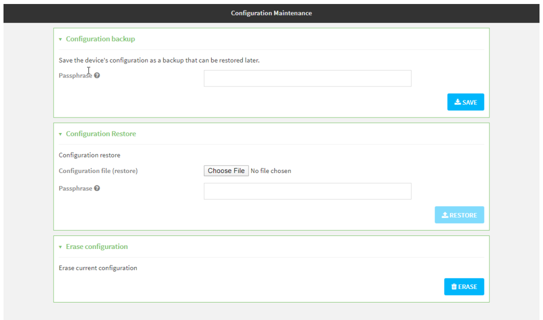
1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. Configure your AnywhereUSB Plus device to match the desired custom factory default configuration.

   For example, you may want to configure the device to use a custom APN or a particular network configuration, so that when you reset the device to factory defaults, it will automatically have your required network configuration.

3. On the main menu, click **System**. Under **Configuration**, click **Configuration Maintenance**.

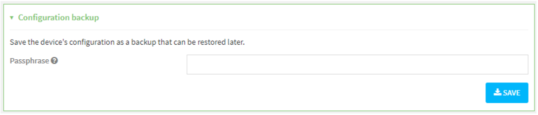


The **Configuration Maintenance** windows is displayed.



Save the device's configuration as a backup that can be restored later.

4. In the **Configuration backup** section, click **SAVE**.



Save the device's configuration as a backup that can be restored later.

Do not set a **Passphrase** for the configuration backup. The file will be downloaded using your browser's standard download process.

5. After the configuration backup file has been downloaded, rename the file to:

   custom-default-config.bin

6. Upload the file to the device, into the **/opt** directory.

   See Upload and download files for information about uploading a file to the device.

   If you use th Web UI to upload the file, you will need to use the **mv** command at the Admin CLI to move the file to the **/opt** directory. For example:

```
> mv /etc/config/scripts/custom-default-config.bin /opt
>
```

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Enter the following:

```
> system backup /opt/custom-default-config.bin type archive
Backup saved as /opt/custom-default-config.bin
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configuration files

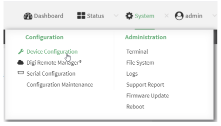
The AnywhereUSB Plus configuration file, /etc/config/accns.json, contains all configuration changes that have been made to the device. It does not contain the complete device configuration; it only contains changes to the default configuration. Both the default configuration and the changes contained in the accns.json file are applied when the device reboots.

Save configuration changes

When you make changes to the AnywhereUSB Plus configuration, the changes are not automatically saved. You must explicitly save configuration changes, which also applies the changes. If you do not save configuration changes, the system discards the changes.

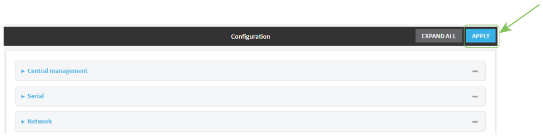
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Make any necessary configuration changes.
4. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Make any necessary configuration changes.
4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

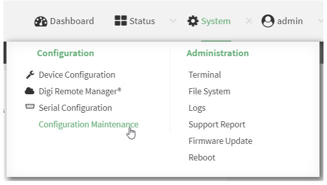
Save configuration to a file

You can save your AnywhereUSB Plus device's configuration to a file and use this file to restore the configuration, either to the same device or to similar devices.

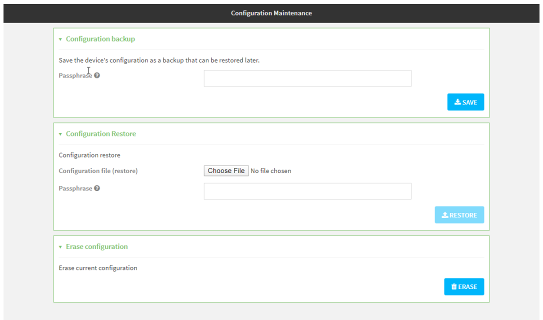
WebUI

This procedure creates a binary archive file containing the device's configuration, certificates and keys, and other information.

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Configuration**, click **Configuration Maintenance**.

   The **Configuration Maintenance** windows is displayed.

   Save the device's configuration as a backup that can be restored later.

3. In the **Configuration backup** section:
   a. (Optional) To encrypt the configuration using a passphrase, for **Passphrase (save/restore)**, enter the passphrase.
   b. Click **SAVE**.

   The file will be downloaded using your browser's standard download process.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Enter the following:

```
> system backup path [passphrase passphrase] type type
```

where

- *path* is the location on the AnywhereUSB Plus's filesystem where the configuration backup file should be saved.
- *passphrase* (optional) is a passphrase used to encrypt the configuration backup.
- *type* is the type of backup, either:
  - **archive**: Creates a binary archive file containing the device's configuration, certificates and keys, and other information.
  - **cli-config**: Creates a text file containing only the configuration changes.

For example:

```
> system backup /etc/config/scripts/ type archive
```

3. (Optional) Use **scp** to copy the file from your device to another host:

```
> scp host hostname-or-ip user username remote remote-path local local-path
to remote
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the location on the remote host where the file will be copied.
- *local-path* is the path and filename on the AnywhereUSB Plus device.

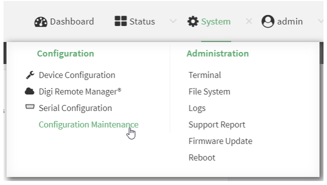
```
> scp host 192.168.4.1 user admin remote /home/admin/bin/ local
/etc/config/backup-archive-0040FF800120-19.05.17-19.01.17.bin to remote
```

Restore the device configuration

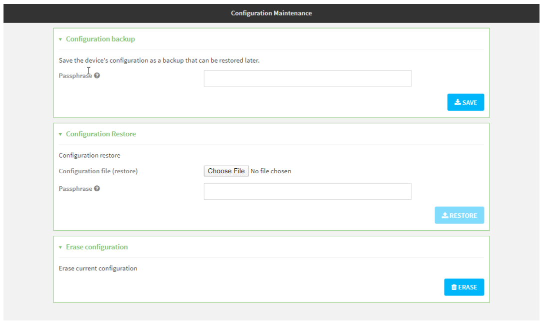
You can restore a configuration file to your AnywhereUSB Plus device by using a backup from the device, or a backup from a similar device.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Configuration**, click **Configuration Maintenance**.



The **Configuration Maintenance** windows is displayed.



Save the device's configuration as a backup that can be restored later.

Passphrase

Configuration restore

Configuration file (restore) No file chosen

Passphrase

Erase current configuration

3. In the **Configuration Restore** section:
   a. If a passphrase was used to create the configuration backup, for **Passphrase (save/restore)**, enter the passphrase.
   b. Under **Configuration Restore**, click **Choose File**.
   c. Browse to the system firmware file location on your local computer and select the file.
   d. Click **RESTORE**.
4. Click **CONFIRM**.

   The configuration will be restored and the device will be rebooted.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. If the configuration backup is on a remote host, use **scp** to copy the file from the host to your device:

```
> scp host hostname-or-ip user username remote remote-path local local-path
to local
```

   where:

   - *hostname-or-ip* is the hostname or ip address of the remote host.
   - *username* is the name of the user on the remote host.
   - *remote-path* is the path and filename of the file on the remote host that will be copied to the AnywhereUSB Plus device.
   - *local-path* is the location on the AnywhereUSB Plus device where the copied file will be placed.

```
> scp host 192.168.4.1 user admin remote /home/admin/bin/backup-archive-
0040FF800120-21.2.39.67-19.23.42.bin local /opt to local
```

3. Enter the following:

```
> system restore filepath [passphrase passphrase]
```

   where

- *filepath* is the the path and filename of the configuration backup file on the AnywhereUSB Plus's filesystem (*local-path* in the previous step).
- *passphrase* (optional) is the passphrase to restore the configuration backup, if a passphrase was used when the backup was created.

For example:

```
> system restore /opt/backup-archive-0040FF800120-21.2.39.67-
19.23.42.bin
```

Schedule system maintenance tasks

You can configure tasks and custom scripts to be run during a specified maintenance window.

Required configuration items

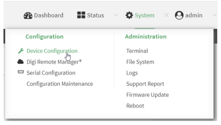
- The time that the system maintenance tasks will start.
- The duration window during which the system maintenance tasks can run.
- The frequency (either daily or weekly) that the tasks will run.
- The tasks to be performed. Options are:
  - Modem firmware update.
  - Configuration check.

Additional configuration items

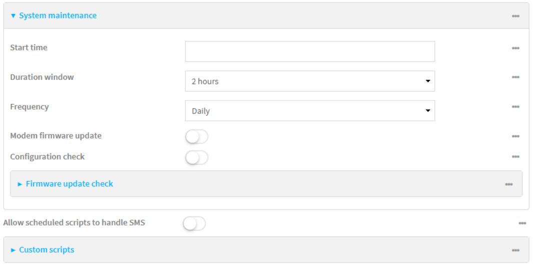
- Custom scripts that should be run as part of the configuration check.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **System** > **Scheduled tasks** > **System maintenance**.

   
   ▼ System maintenance
   Start time
   Duration window: 2 hours
   Frequency: Daily
   Modem firmware update
   Configuration check
   ► Firmware update check
   Allow scheduled scripts to handle SMS
   ► Custom scripts

4. For **Start time**, type the time of day that the maintenance window should start, using the syntax *HH*:*MM*. If **Start time** is not set, maintenance tasks are not scheduled and will not be run.

   The behavior of **Start time** varies depending on the setting of **Duration window**, which is configured in the next step.

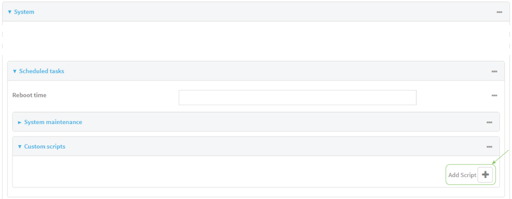
   - If **Duration window** is set to **Immediately**, all scheduled tasks will begin at the exact time specified in **Start time**.

  - If **Duration window** is set to **24 hours**, **Start time** is effectively obsolete and the maintenance tasks will be scheduled to run at any time. Setting **Duration window** to **24 hours** can potentially overstress the device and should be used with caution.
  - If **Duration window** is set to any value other than to **Immediately** or **24 hours**, the maintenance tasks will run at a random time during the time allotted for the duration window.
  - If **Duration window** is set to one or more hours, the minutes field in **Start time** is ignored and the duration window will begin at the beginning of the specified hour.

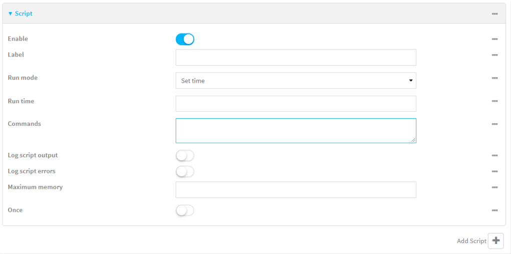
5. For **Duration window**, select the amount of time that the maintenance tasks will be run. If **Immediately** is selected, all scheduled tasks will begin at the exact time specified in **Start time**.
6. For **Frequency**, select either **Daily** or **Weekly** for the frequency that the maintenance tasks should be run.
7. (Optional) Click to enable **Modem firmware update** to instruct the system to look for any updated modem firmware during the maintenance window. If updated firmware is found, it will then be installed. Modem firmware update looks for updated firmware both on the local device and over the network, using either a WAN or cellular connection.
8. (Optional) Click to enable **Configuration check** to allow for the configuration to be updated, including by custom scripts, during the maintenance window.
9. (Optional) Configure automated checking for device firmware updates:
    a. Click to expand **Firmware update check**.
    b. **Device firmware update check** is enabled by default. This enables to automated checking for device firmware updates.
    c. **Modem firmware update check** is enabled by default. This enables to automated checking for modem firmware updates.
    d. For Frequency, select how often automated checking for device and modem firmware should take place. Allowed values are **Daily**, **Weekly**, and **Monthly**. The default is **Daily**.
10. (Optional) Enable **Allow scheduled scripts to handle SMS** to allow scheduled scripts to handle SMS messages.
11. (Optional) To schedule custom scripts:
    a. Click **Custom scripts**.

    Note This feature does not provide syntax or error checking. Certain commands can render the device inoperable. Use with care. Scripts created here are also automatically entered in **Configuration** > **Applications**.

    b. For **Add Script**, click ✚.



The schedule script configuration window is displayed.



▾ Script

Enable ●

Label

Run mode: Set time

Run time

Commands

Log script output ○

Log script errors ○

Maximum memory

Once ○

Add Script ➕

Scheduled scripts are enabled by default. To disable, click **Enable** to toggle off.

c. (Optional) For **Label**, provide a label for the script.

d. For **Run mode**, select the mode that will be used to run the script. Available options are:

   - **On boot**: The script will run once each time the device boots.
     - If **On boot** is selected, select the action that will be taken when the script completes in **Exit action**. Available options are:
       - **None**: Action taken when the script exits.
       - **Restart script**: Runs the script repeatedly.
       - **Reboot**: The device will reboot when the script completes.
   - **Interval**: The script will start running at the specified interval, within 30 seconds after the configuration change is saved.
     - If **Interval** is selected, in **Interval**, type the interval.

       Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

       For example, to set **Interval** to ten minutes, enter **10m** or **600s**.
     - Click to enable **Run single** to run only a single instance of the script at a time.

       If **Run single** is not selected, a new instance of the script will be started at every interval, regardless of whether the script is still running from a previous interval.
   - **Set time**: Runs the script at a specified time of the day.
     - If **Set Time** is selected, specify the time that the script should run in **Run time**, using the format *HH:MM*.
   - **During system maintenance**: The script will run during the system maintenance time window.

e. For **Commands**, enter the commands that will execute the script.

   If the script begins with **#!**, then the script will be invoked in the location specified by the path for the script command. Otherwise, the default shell will be used (equivalent to **#!/bin/sh**).

f. Script logging options:

   i. Click to enable **Log script output** to log the script's output to the system log.

   ii. Click to enable **Log script errors** to log script errors to the system log.

   If neither option is selected, only the script's exit code is written to the system log.

   g. For **Maximum memory**, enter the maximum amount of memory available to be used by the script and its subprocesses, using the format *number* {**b**|**bytes**|**KB**|**k**|**MB**|**MB**|**M**|**GB**|**G**|**TB**|**T**}.

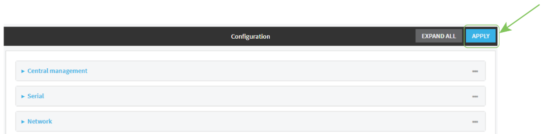
   h. Click to enable **Once** to configure the script to run only once at the specified time.

      If **Once** is enabled, rebooting the device will cause the script to not run again. The only way to re-run the script is to:

      - Remove the script from the device and add it again.
      - Make a change to the script.
      - Uncheck **Once**.

   i. **Sandbox** is enabled by default, which restricts access to the file system and available commands that can be used by the script. This option protects the script from accidentally destroying the system it is running on.

12. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Schedule system maintenance:

   a. Configure the time of day that the maintenance window should start, using the syntax *HH*:*MM*. If the start time is not set, maintenance tasks are not scheduled and will not be run.

```
(config)> system schedule maintenance from HH:MM
(config)>
```

   The behavior of the start time varies depending on the setting of the duration length, which is configured in the next step.

   - If the duration length is set to **0**, all scheduled tasks will begin at the exact time specified in the start time.
   - If the duration length is set to **24 hours**, the start time is effectively obsolete and the maintenance tasks will be scheduled to run at any time. Setting the duration length to **24 hours** can potentially overstress the device and should be used with caution.

- If the duration length is set to any value other than to **0** or **24 hours**, the maintenance tasks will run at a random time during the time allotted for the duration window.
- If the duration length is set to one or more hours, the minutes field in the start time is ignored and the duration window will begin at the beginning of the specified hour.

b. Configure the duration length (the amount of time that the maintenance tasks will be run). If **0** is used, all scheduled tasks will begin at the start time, defined in the previous step.

```
(config)> system schedule maintenance length num
(config)>
```

where *num* is any whole number between **0** and **24**.

c. Configure the frequency that the maintenance tasks should be run:

```
(config)> system schedule maintenance frequency value
(config)>
```

where *value* is either **daily** or **weekly**. **Daily** is the default.

4. Configure the device to look for any updated modem firmware during the maintenance window. If updated firmware is found, it will then be installed. The device will look for updated firmware both on the local device and over the network, using either a WAN or cellular connection.

```
system schedule maintenance modem_fw_update value
(config)>
```

where *value* is either **true** or **false**. **yes** or **no**, and **1** or **0** are also allowed.

5. (Optional) Configure automated checking for device firmware updates:

a. **Device firmware update check** is enabled by default. This enables to automated checking for device firmware updates. To disable:

```
(config)> system schedule maintenance firmware_update_check device false
(config)>
```

b. Set how often automated checking for device firmware should take place:

```
(config)> system schedule maintenance frequency value
(config)>
```

where *value* is either **daily**, **weekly**, or **monthly**. **daily** is the default.

6. (Optional) Allow scheduled scripts to handle SMS messages:

```
(config)> system schedule sms_script_handling true
(config)>
```

7. (Optional) Schedule custom scripts:

a. Add a script:

```
(config)> add system schedule script end
(config system schedule script 0)>
```

Scheduled scripts are enabled by default. To disable:

```
(config system schedule script 0)> enable false
(config system schedule script 0)>
```

b. (Optional) Provide a label for the script.

```
(config system schedule script 0)> label value
(config system schedule script 0)>
```

where *value* is any string. if spaces are used, enclose *value* within double quotes.

c. Set the mode that will be used to run the script:

```
(config system schedule script 0)> when mode
(config system schedule script 0)>
```

where *mode* is one of the following:

- **boot**: The script will run once each time the device boots.
    - If **boot** is selected, set the action that will be taken when the script completes:

      ```
      (config system schedule script 0)> exit_action action
      (config system schedule script 0)>
      ```

      where *action* is one of the following:

        - **none**: Action taken when the script exits.
        - **restart**: Runs the script repeatedly.
        - **reboot**: The device will reboot when the script completes.
- **interval**: The script will start running at the specified interval, within 30 seconds after the configuration change is saved. If **interval** is selected:
    - Set the interval:

      ```
      (config system schedule script 0)> on_interval value
      (config system schedule script 0)>
      ```

      where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

      For example, to set **on_interval** to ten minutes, enter either **10m** or **600s**:

      ```
      (config system schedule script 0)> on_interval  600s
      (config system schedule script 0)>
      ```
    - (Optional) Configure the script to run only a single instance at a time:

      ```
      (config system schedule script 0)> once true
      (config system schedule script 0)>
      ```

      If **once** is set to **false**, a new instance of the script will be started at every interval, regardless of whether the script is still running from a previous interval.

- **set_time**: Runs the script at a specified time of the day.
    - If **set_time** is set, set the time that the script should run, using the format *HH*:*MM*:

      ```
      (config system schedule script 0)> run_time HH:MM
      (config system schedule script 0)>
      ```

- **maintenance_time**: The script will run during the system maintenance time window.

d. Set the commands that will execute the script:

```
(config system schedule script 0)> commands filename
(config system schedule script 0)>
```

where *filename* is the path and filename of the script, and any related command line information.

If the script begins with **#!**, then the script will be invoked in the location specified by the path for the script command. Otherwise, the default shell will be used (equivalent to **#!/bin/sh**).

e. Script logging options:

- To log the script's output to the system log:

  ```
  (config system schedule script 0)> syslog_stdout true
  (config system schedule script 0)>
  ```

- To log script errors to the system log:

  ```
  (config system schedule script 0)> syslog_stderr true
  (config system schedule script 0)>
  ```

If **syslog_stdout** and **syslog_stderr** are not enabled, only the script's exit code is written to the system log.

f. Set the maximum amount of memory available to be used by the script and its subprocesses:

```
(config system schedule script 0)> max_memory value
(config system schedule script 0)>
```

where *value* uses the syntax ***number***{**b**|**bytes**|**KB**|**k**|**MB**|**MB**|**M**|**GB**|**G**|**TB**|**T**}.

g. To run the script only once at the specified time:

```
(config system schedule script 0)> once true
(config system schedule script 0)>
```

If **once** is enabled, rebooting the device will cause the script to run again. The only way to re-run the script is to:

- Remove the script from the device and add it again.
- Make a change to the script.
- Disable **once**.

h. **Sandbox** is enabled by default. This option protects the script from accidentally destroying the system it is running on.

```
(config system schedule script 0)> sandbox true
(config system schedule script 0)>
```

8. Allow for the configuration to be updated, including by custom scripts, during the maintenance window:

```
system schedule maintenance config_check value
(config)>
```

   where *value* is either **true** or **false**. **yes** or **no**, and **1** or **0** are also allowed.

9. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

10. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Disable device encryption

You can disable the cryptography on your AnywhereUSB Plus device. This can be used to ship unused devices from overseas without needing export licenses from the country from which the device is being shipped.

When device encryption is disabled, the following occurs:

- The device is reset to the default configuration and rebooted.
- After the reboot:
  - Access to the device via the WebUI and SSH are disabled.
  - All internet connectivity is disabled, including WAN and WWAN. Connectivity to central management software is also disabled.
  - All IP networks and addresses are disabled except for the default 192.168.210.1/24 network on the local LAN Ethernet port. DHCP server is also disabled.

    The device can only be accessed by using telnet from a local machine connecting to the 192.168.210.1/24 network.

Disabling device encryption is not available in the WebUI. It can only be performed from the Admin CLI.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Disable encryption with the following command:

```
> system disable-cryptography
>
```

3. Type **exit** to exit the Admin CLI.

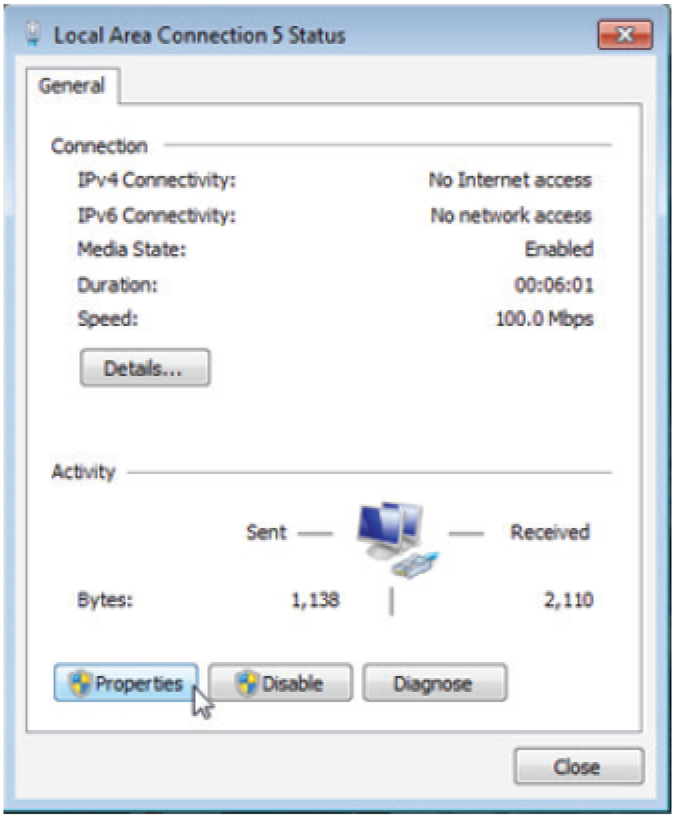
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Re-enable cryptography after it has been disabled.

To re-enable cryptography:

1. Configure your PC network to connect to the 192.168.210 subnet. For example, on a Windows PC:

   a. Select the **Properties** of the relevant network connection on the Windows PC.



Local Area Connection 5 Status

Connection

IPv4 Connectivity:	No Internet access
IPv6 Connectivity:	No network access
Media State:	Enabled
Duration:	00:06:01
Speed:	100.0 Mbps

Activity

	Sent	Received
Bytes:	1,138	2,110

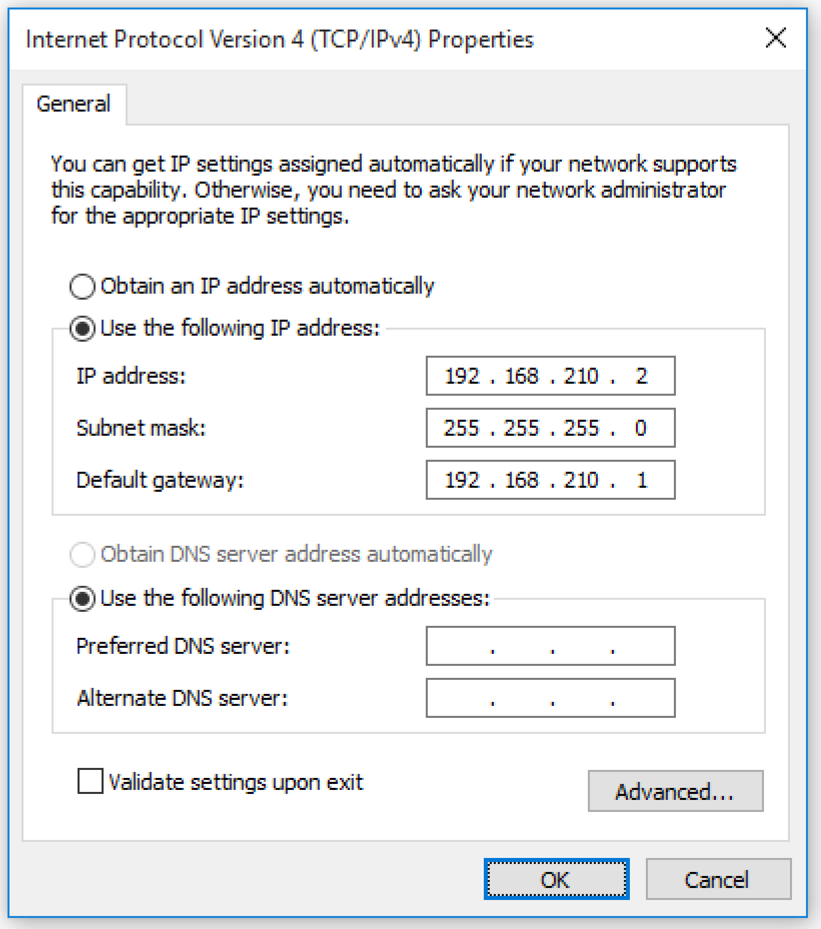
   b. Click the **Internet Protocol Version 4 (TCP/IPv4)** parameter.

   c. Click **Properties**. The **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog appears.

   d. Configure with the following details:

      - **IP address** for PC: 192.168.210.2
      - **Subnet**: 255.255.255.0

- **Gateway**: 192.168.210.1



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 210 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 210 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

2. Connect the PC's Ethernet port to the ETH1 Ethernet port on your AnywhereUSB Plus device.
3. Open a telnet session and connect to the AnywhereUSB Plus device at the IP address of 192.168.210.1.
4. Log into the device:
   - Username: **admin**
   - Password: The default unique password for your device is printed on the device label.
5. At the shell prompt, type:

```
# rm /etc/config/.nocrypt
# flatfsd -i
```

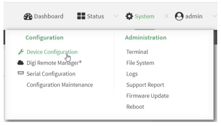
This will re-enable encryption and leave the device at its factory default setting.

Configure the speed of your Ethernet ports

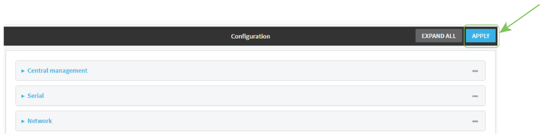
You can configure the speed of your AnywhereUSB Plus device's Ethernet ports.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Network** > **Device**.
4. Click to expand the Ethernet port to be configured.
5. For **Speed**, select the appropriate speed for the Ethernet port, or select **Auto** to automatically detect the speed. The default is **Auto**.
6. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. At the config prompt, type:

   ```
   (config)> network device eth_port value
   ```

   where:

   - *eth_port* is the name of the Ethernet port (for example, **eth1**)
   - *value* is one of:
     - **10**—Sets the speed to 10 Mbps.
     - **100**—Sets the speed to 100 Mbps.
     - **1000**—Sets the speed to 1 Gbps. Available only for devices with Gigabit Ethernet ports.

       auto—Configures the device to automatically determine the best speed for the Ethernet port.

   The default is **auto**.
4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Monitoring

This chapter contains the following topics:

intelliFlow

intelliFlow monitors system information, network data usage, and traffic information, and displays the information in a series of charts available in the local WebUI. To use intelliFlow, the AnywhereUSB Plus must be powered on and you must have access to the local WebUI. Once you enable intelliFlow, the **Status** > **intelliFlow** option is available in the main menu. By default, intelliFlow is disabled.

intelliFlow provides charts on the following information:

- System utilisation
- Top data usage by host
- Top data usage by server
- Top data usage by service
- Host data usage over time

intelliFlow charts are dymanic; at any point, you can click inside the chart to drill down to view more granular information, and menu options allow you to change various aspects of the information being displayed.

Note When intelliFlow is enabled, it adds an estimated 50MB of data usage for the device by reporting the metrics to Digi Remote Manager.

Enable intelliFlow

Required configuration items

- Enable intelliFlow.

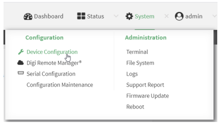
Additional configuration items

- The firewall zone for internal clients being monitored by intelliFlow.

To enable intelliFlow:

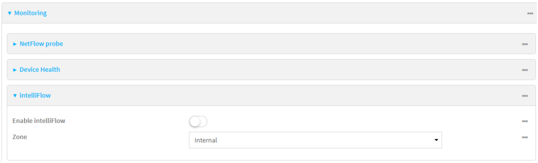
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Monitoring** > **intelliFlow**.

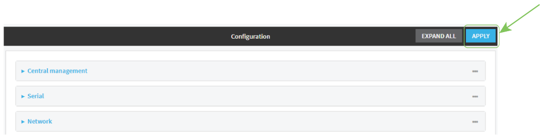
   The intelliFlow configuration window is displayed.



Enable intelliFlow ○

Zone: Internal

4. Click **Enable intelliFlow**.
5. For **Zone**, select the firewall zone. Internal clients that are being monitored by IntelliFlow should be present on the specified zone.
6. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable IntelliFlow:

```
(config)> monitoring intelliflow enable true
```

4. Set the firewall zone. Internal clients that are being monitored by IntelliFlow should be present on the specified zone:
   a. Determine available zones:

```
(config)> monitoring intelliflow zone ?

Zone: The firewall zone which is assigned to the network interface(s)
that
intelliFlow will see as internal clients.  intelliFlow relies on an
internal to
external relationship, where the internal clients are present on the
zone specified.
Format:
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup
Default value: internal
Current value: internal

(config)>
```

b. Set the zone to be used by IntelliFlow:

```
(config)> monitoring intelliflow zone my_zone
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Use intelliFlow to display average CPU and RAM usage

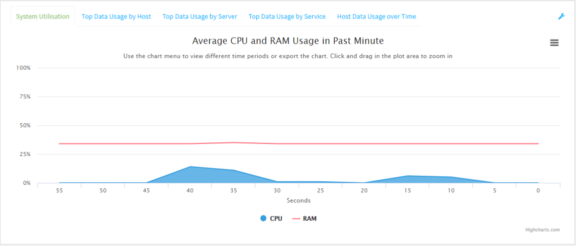
This procedure is only available from the WebUI.

To display display average CPU and RAM usage:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. If you have not already done so, enable intelliFlow. See Enable intelliFlow.
3. From the menu, click **Status** > **intelliFlow**.

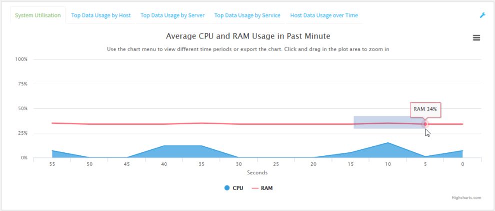
The System Utilisation chart is displayed:



Average CPU and RAM Usage in Past Minute

Use the chart menu to view different time periods or export the chart. Click and drag in the plot area to zoom in

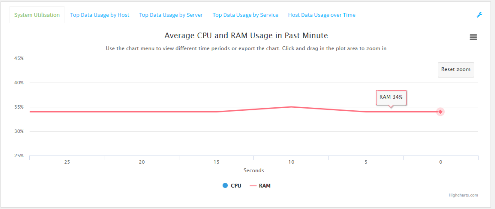
- Display more granular information:
  1. Click and drag over an area in the chart to zoom into that area and provide more granular information.



Average CPU and RAM Usage in Past Minute

Use the chart menu to view different time periods or export the chart. Click and drag in the plot area to zoom in

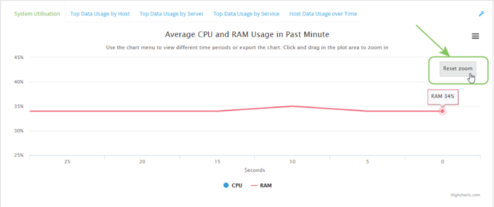
  2. Release to display the selected portion of the chart:



Average CPU and RAM Usage in Past Minute

Use the chart menu to view different time periods or export the chart. Click and drag in the plot area to zoom in

3. Click **Reset zoom** to return to the original display:



Average CPU and RAM Usage in Past Minute

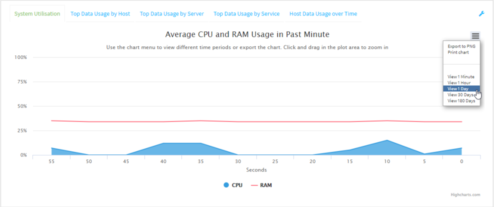
Use the chart menu to view different time periods or export the chart. Click and drag in the plot area to zoom in

- Change the time period displayed by the chart.

  By default, the **System utilisation** chart displays the average CPU and RAM usage over the last minute. You can change this to display the average CPU and RAM usage:

  - Over the last hour.
  - Over the last day.
  - Over the last 30 days.
  - Over the last 180 days.

  1. Click the menu icon (☰).
  2. Select the time period to be displayed.



Average CPU and RAM Usage in Past Minute

Use the chart menu to view different time periods or export the chart. Click and drag in the plot area to zoom in

- Save or print the chart.
  1. Click the menu icon (☰).
  2. To save the chart to your local filesystem, select **Export to PNG**.
  3. To print the chart, select **Print chart**.

Use intelliFlow to display top data usage information

With intelliFlow, you can display top data usage information based on the following:

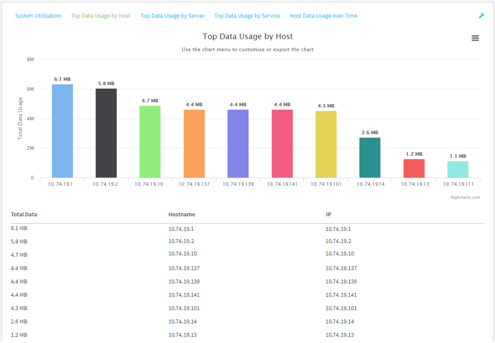
- Top data usage by host
- Top data usage by server
- Top data usage by service

To generate a top data usage chart:

☰ WebUI

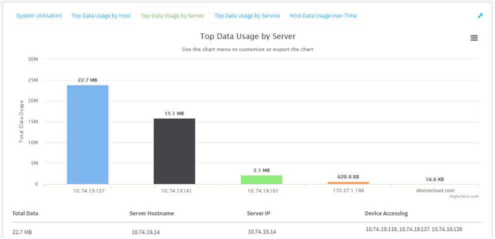
1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. If you have not already done so, enable intelliFlow. See Enable intelliFlow.
3. From the menu, click **Status** > **intelliFlow**.

4. Display a data usage chart:
   - To display the **Top Data Usage by Host** chart, click **Top Data Usage by Host**.



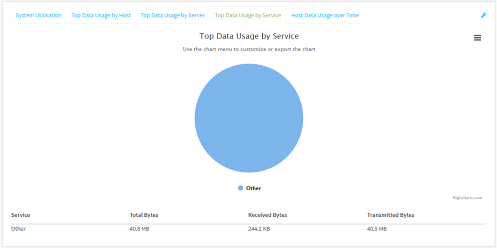
Total Data	Hostname	IP
6.1 MB	10.74.19.1	10.74.19.1
5.8 MB	10.74.19.2	10.74.19.2
4.7 MB	10.74.19.10	10.74.19.10
4.4 MB	10.74.19.137	10.74.19.137
4.4 MB	10.74.19.139	10.74.19.139
4.4 MB	10.74.19.141	10.74.19.141
4.3 MB	10.74.19.101	10.74.19.101
2.6 MB	10.74.19.14	10.74.19.14
1.2 MB	10.74.19.13	10.74.19.13

   - To display the **Top Data Usage by Server** chart, click **Top Data Usage by Server**.



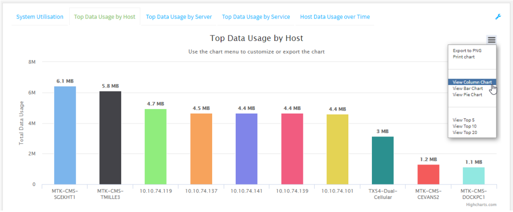
Total Data	Server Hostname	Server IP	Device Accessing
22.7 MB	10.74.19.14	10.74.19.14	10.74.19.119, 10.74.19.137, 10.74.19.139

   - To display the **Top Data Usage by Service** chart, click **Top Data Usage by Service**.


System Utilisation
Top Data Usage by Host
Top Data Usage by Server
Top Data Usage by Service
Host Data Usage over Time
Top Data Usage by Service
Use the chart menu to customize or export the chart
Other
Highcharts.com

Service	Total Bytes	Received Bytes	Transmitted Bytes
Other	40.8 MB	244.2 KB	40.5 MB

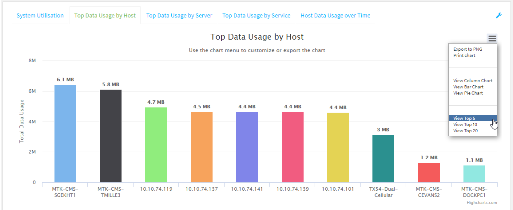
5. Change the type of chart that is used to display the data:
   a. Click the menu icon (☰).
   b. Select the type of chart.



6. Change the number of top users displayed.

   You can display the top five, top ten, or top twenty data users.

  a. Click the menu icon (☰).
  b. Select the number of top users to displayed.



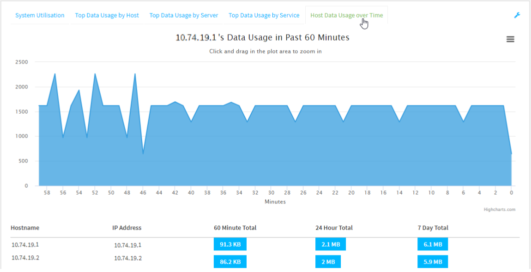
7. Save or print the chart.
   a. Click the menu icon (☰).
   b. To save the chart to your local filesystem, select **Export to PNG**.
   c. To print the chart, select **Print chart**.

Use intelliFlow to display data usage by host over time

To generate a chart displaying a host's data usage over time:

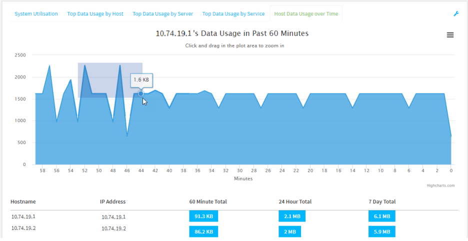
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. If you have not already done so, enable intelliFlow. See Enable intelliFlow.
3. From the menu, click **Status** > **intelliFlow**.
4. Click **Host Data Usage Over Time**.



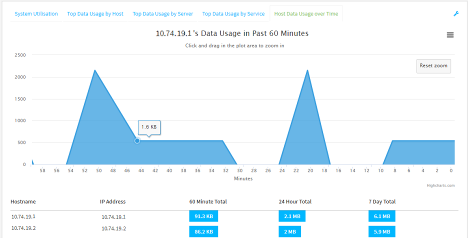
Hostname	IP Address	60 Minute Total	24 Hour Total	7 Day Total
10.74.19.1	10.74.19.1	91.3 KB	2.1 MB	6.1 MB
10.74.19.2	10.74.19.2	86.2 KB	2 MB	5.9 MB

- Display more granular information:
  a. Click and drag over an area in the chart to zoom into that area and provide more granular information.



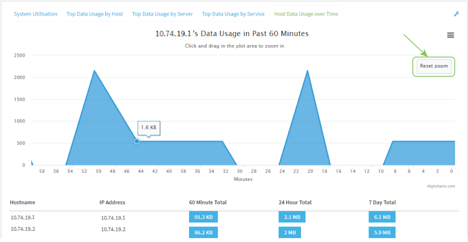
Hostname	IP Address	60 Minute Total	24 Hour Total	7 Day Total
10.74.19.1	10.74.19.1	91.3 KB	2.1 MB	6.1 MB
10.74.19.2	10.74.19.2	86.2 KB	2 MB	5.9 MB

b. Release to display the selected portion of the chart:



Hostname	IP Address	60 Minute Total	24 Hour Total	7 Day Total
10.74.19.1	10.74.19.1	91.3 KB	2.1 MB	6.1 MB
10.74.19.2	10.74.19.2	86.2 KB	2 MB	5.9 MB

c. Click **Reset zoom** to return to the original display:



Hostname	IP Address	60 Minute Total	24 Hour Total	7 Day Total
10.74.19.1	10.74.19.1	91.3 KB	2.1 MB	6.1 MB
10.74.19.2	10.74.19.2	86.2 KB	2 MB	5.9 MB

- Save or print the chart.
  a. Click the menu icon (☰).
  b. To save the chart to your local filesystem, select **Export to PNG**.
  c. To print the chart, select **Print chart**.

Configure NetFlow Probe

NetFlow probe is used to probe network traffic on the AnywhereUSB Plus device and export statistics to NetFlow collectors.

Required configuration items

- Enable NetFlow.
- The IP address of a NetFlow collector.

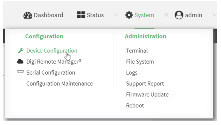
Additional configuration items

- The NetFlow version.
- Enable flow sampling and select the flow sampling technique.
- The number of flows from which the flow sampler can sample.
- The number of seconds that a flow is inactive before it is exported to the NetFlow collectors.
- The number of seconds that a flow is active before it is exported to the NetFlow collectors.
- The maximum number of simultaneous flows.
- A label for the NetFlow collector.
- The port of the NetFlow collector.
- Additional NetFlow collectors.

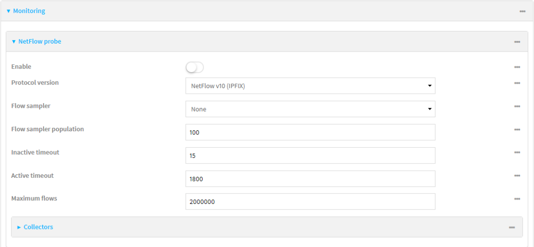
To probe network traffic and export statistics to NetFlow collectors:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Monitoring** > **NetFlow probe**.

   ▾ Monitoring

   ▾ NetFlow probe

   Enable

   Protocol version: NetFlow v10 (IPFIX)

   Flow sampler: None

   Flow sampler population: 100

   Inactive timeout: 15

   Active timeout: 1800

   Maximum flows: 2000000

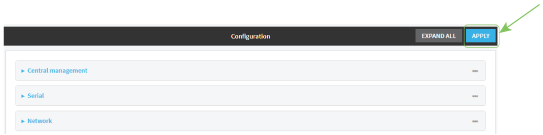
   ▸ Collectors

4. **Enable** NetFlow probe.
5. **Protocol version:** Select the **Protocol version**. Available options are:
   - **NetFlow v5**—Supports IPv4 only.
   - **NetFlow v9**—Supports IPv4 and IPv6.
   - **NetFlow v10 (IPFIX)**—Supports both IPv4 and IPv6 and includes IP Flow Information Export (IPFIX).

   The default is **NetFlow v10 (IPFIX)**.
6. Enable **Flow sampler** by selecting a sampling technique. Flow sampling can reduce flow processing and transmission overhead by providing a representative subset of all flows. Available options are:
   - **None**—No flow sampling method is used. Each flow is accounted.
   - **Deterministic**—Selects every *n*th flow, where *n* is the value of **Flow sampler population**.
   - **Random**—Randomly selects one out of every *n* flows, where *n* is the value of **Flow sampler population**.
   - **Hash**—Randomly selects one out of every *n* flows using the hash of the flow key, where *n* is the value of **Flow sampler population**.
7. For **Flow sampler population**, if you selected a flow sampler, enter the number of flows for the sampler. Allowed value is any number between **2** and **16383**. The default is **100**.
8. For **Inactive timeout**, type the the number of seconds that a flow can be inactive before sent to a collector. Allowed value is any number between **1** and **15**. The default is **15**.
9. For **Active timeout**, type the number of seconds that a flow can be active before sent to a collector. Allowed value is any number between **1** and **1800**. The default is **1800**.
10. For **Maximum flows**, type the maximum number of flows to probe simultaneously. Allowed value is any number between **0** and **2000000**. The default is **2000000**.

11. Add collectors:
    a. Click to expand **Collectors**.
    b. For **Add Collector**, click ✚.
    c. (Optional) Type a **Label** for the collector.
    d. For **Address**, type the IP address of the collector.
    e. (Optional) For **Port**, enter the port number used by the collector. The default is 2055.

    Repeat to add additional collectors.
12. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable NetFlow:

```
(config)> monitoring netflow enable true
(config)>
```

4. Set the protocol version:

```
(config)> monitoring netflow protocol version
                                                (config)>
```

   where *version* is one of:

   - **v5**—NetFlow v5 supports IPv4 only.
   - **v9**—NetFlow v9 supports IPv4 and IPv6.
   - **v10**—NetFlow v10 (IPFIX) supports both IPv4 and IPv6 and includes IP Flow Information Export (IPFIX).

   The default is **v10**.

1. Enable flow sampling by selecting a sampling technique. Flow sampling can reduce flow processing and transmission overhead by providing a representative subset of all flows.

```
(config)> monitoring netflow sampler type
(config)>
```

where *type* is one of:

- **none**—No flow sampling method is used. Each flow is accounted.
- **deterministic**—Selects every *n*th flow, where *n* is the value of the flow sample population.
- **random**—Randomly selects one out of every *n* flows, where *n* is the value of the flow sample population.
- **hash**—Randomly selects one out of every *n* flows using the hash of the flow key, where *n* is the value of the flow sample population.

5. If you are using a flow sampler, set the number of flows for the sampler:

```
(config)> monitoring netflow sampler_population value
(config)>
```

where *value* is any number between **2** and **16383**. The default is **100**.

6. Set the number of seconds that a flow can be inactive before sent to a collector:

```
(config)> monitoring netflow inactive_timeout value
(config)>
```

where *value* is any is any number between **1** and **15**. The default is **15**.

7. Set the number of seconds that a flow can be active before sent to a collector:

```
(config)> monitoring netflow active_timeout value
(config)>
```

where *value* is any is any number between **1** and **1800**. The default is **1800**.

8. Set the maximum number of flows to probe simultaneously:

```
(config)> monitoring netflow max_flows value
(config)>
```

where *value* is any is any number between **0** and **2000000**. The default is **2000000**.

9. Add collectors:

   a. Add a collector:

   ```
   (config)> add monitoring netflow collector end
   (config monitoring netflow collector 0)>
   ```

   b. Set the IP address of the collector:

   ```
   (config monitoring netflow collector 0)> address ip_address
   (config monitoring netflow collector 0)>
   ```

   c. (Optional) Set the port used by the collector:

   ```
   (config monitoring netflow collector 0)> port port
   (config monitoring netflow collector 0)>
   ```

d. (Optional) Set a label for the collector:

```
(config monitoring netflow collector 0)> label "This is a collector."
(config monitoring netflow collector 0)>
```

Repeat to add additional collectors.

10. Save the configuration and apply the change:

```
(config monitoring netflow collector 0)> save
Configuration saved.
>
```

11. Type **exit** to exit the Admin CLI.

Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Central management

This chapter contains the following topics:

Digi Remote Manager support

Digi Remote Manager is a hosted remote configuration and management system that allows you to remotely manage a large number of devices. Remote Manager includes a web-based interface that you can use to perform device operations, such as viewing and changing device configurations and performing firmware updates. Remote Manager servers also provide a data storage facility. The Digi Remote Manager is the default cloud-based management system, and is enabled by default. You can also select to use Digi aView as the cloud-based management system. See *Digi aView User Guide* for information about aView.

To use Remote Manager, you must set up a Remote Manager account. To set up a Remote Manager account and learn more about Digi Remote Manager, go to www.digi.com/products/cloud/digi-remote-manager.

To learn more about Remote Manager features and functions, see the *Digi Remote Manager User Guide*.

Configure Digi Remote Manager

By default, your AnywhereUSB Plus device is configured to use central management using Digi Remote Manager.

Additional configuration options

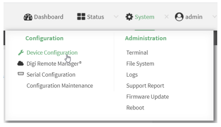
These additional configuration settings are not typically configured, but you can set them as needed:

- Disable the Digi Remote Manager connection if it is not required. You can also configure an alternate cloud-based central management application.
- Change the reconnection timer.
- The non-cellular keepalive timeout.
- The cellular keepalive timeout.
- The keepalive count before the Remote Manager connection is dropped.
- SMS support.
- HTTP proxy server support.

To configure Digi Remote Manager:

WebUI

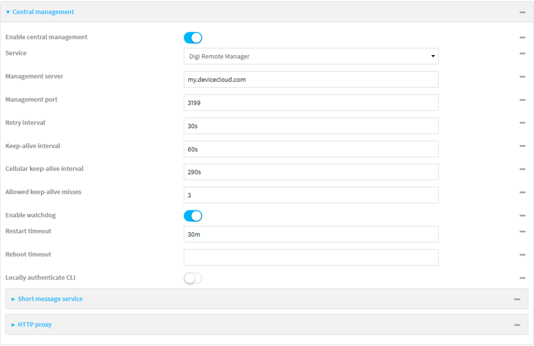
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **Central management**.

   The Central management configuration window is displayed.

   ▼ Central management

Enable central management	●
Service	Digi Remote Manager ▾
Management server	my.devicecloud.com
Management port	3199
Retry interval	30s
Keep-alive interval	60s
Cellular keep-alive interval	290s
Allowed keep-alive misses	3
Enable watchdog	●
Restart timeout	30m
Reboot timeout	
Locally authenticate CLI	○

   ► Short message service

   ► HTTP proxy

   Digi Remote Manager support is enabled by default. To disable, click **Enable central management**.

4. (Optional) For **Service**, select either **Digi Remote Manager** or **Digi aView**. The default is **Digi Remote Manager**.
5. (Optional) For **Management server**, type the URL for the central management server. The default is the Digi Remote Manager server, my.devicecloud.com.
6. (Optional) For **Management port**, type the destination port for the remote cloud services connection. The default is **3199**.
7. (Optional) For **Retry interval**, type the amount of time that the AnywhereUSB Plus device should wait before reattempting to connect to remote cloud services after being disconnected. The default is 30 seconds.

   Allowed values are any number of hours, minutes, or seconds, and take the format ***number*** {**h**|**m**|**s**}.

   For example, to set **Retry interval** to ten minutes, enter **10m** or **600s**.
8. (Optional) For **Keep-alive interval**, type the amount of time that the AnywhereUSB Plus device should wait between sending keep-alive messages to remote cloud services when using a non-cellular interface. The default is 60 seconds.

   Allowed values are any number of hours, minutes, or seconds, and take the format ***number*** {**h**|**m**|**s**}.

   For example, to set **Keep-alive interval** to ten minutes, enter **10m** or **600s**.
9. (Optional) For **Cellular keep-alive interval**, type the amount of time that the AnywhereUSB Plus device should wait between sending keep-alive messages to remote cloud services when using a cellular interface. The default is 290 seconds.

   Allowed values are any number of hours, minutes, or seconds, and take the format ***number*** {**h**|**m**|**s**}.

   For example, to set **Cellular keep-alive interval** to ten minutes, enter **10m** or **600s**.
10. (Optional) For **Allowed keep-alive misses**, type the number of allowed keep-alive misses. The default is **3**.
11. **Enable watchdog** is used to monitor the connection to remote cloud services. If the connection is down, you can configure the device to restart the connection, or to reboot. The watchdog is enabled by default.

12. If **Enable watchdog** is enabled:
    a. (Optional) For **Restart Timeout**, type the amount of time to wait before restarting the connection to the remote cloud services, once the connection is down.

       Allowed values are any number of hours, minutes, or seconds, and take the format ***number*** {**h**|**m**|**s**}.

       For example, to set **Restart Timeout** to ten minutes, enter **10m** or **600s**.

       The minimum value is 30 minutes and the maximum is 48 hours. If not set, this option is disabled. The default is 30 minutes.

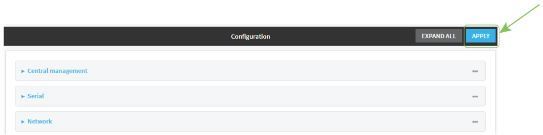
    b. (Optional) For **Reboot Timeout**, type the amount of time to wait before rebooting the device, once the connection to the remote cloud servicesis down. By default, this option is not set, which means that the option is disabled.

       Allowed values are any number of hours, minutes, or seconds, and take the format ***number*** {**h**|**m**|**s**}.

       For example, to set **Reboot Timeout** to ten minutes, enter **10m** or **600s**.

       The minimum value is 30 minutes and the maximum is 48 hours. If not set, this option is disabled. The default is disabled.

13. (Optional) Enable **Locally authenticate CLI** to require a login and password to authenticate the user from the remote cloud services CLI. If disabled, no login prompt will be presented and the user will be logged in as **admin**. The default is disabled.
14. (Optional) Configure the AnywhereUSB Plus device to communicate with remote cloud services by using SMS:
    a. Click to expand **Short message service**.
    b. **Enable** SMS messaging.
    c. For **Destination phone number**, type the phone number for the remote cloud services.
    d. (Optional) Type the **Service identifier**.
15. (Optional) Configure the AnywhereUSB Plus device to communicate with remote cloud services by using an HTTP proxy server:
    a. Click to expand **HTTP Proxy**.
    b. **Enable** the use of an HTTP proxy server.
    c. For **Server**, type the hostname of the HTTP proxy server.
    d. For **Port**, type or select the port number on the HTTP proxy server that the device should connect to. The default is **2138**.
16. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Digi Remote Manager support is enabled by default. To disable Digi Remote Manager support:

   ```
   (config)> cloud enable false
   (config)>
   ```

4. (Optional) Set the service:

   ```
   (config)> cloud service value
   (config)>
   ```

   where *value* is either:

   - **drm**: Digi Remote Manager
   - **aview**: Digi aView

   The default is Digi Remote Manager.

5. (Optional) Set the URL for the central management server. The default is the Digi Remote Manager server, my.devicecloud.com.

   ```
   (config)> cloud drm drm_url url
   (config)>
   ```

6. (Optional) Set the amount of time that the AnywhereUSB Plus device should wait before reattempting to connect to the remote cloud services after being disconnected. The minimum value is ten seconds. The default is 30 seconds.

   ```
   (config)> cloud drm retry_interval value
   ```

   where *value* is any number of hours, minutes, or seconds, and takes the format ***number***{**h**|**m**|**s**}.

   For example, to set **the retry interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config)> cloud drm retry_interval 600s
   (config)>
   ```

7. (Optional) Set the amount of time that the AnywhereUSB Plus device should wait between sending keep-alive messages to the Digi Remote Manager when using a non-cellular interface. Allowed values are from 30 seconds to two hours. The default is 60 seconds.

   ```
   (config)> cloud drm keep_alive value
   (config)>
   ```

   where *value* is any number of hours, minutes, or seconds, and takes the format ***number***{**h**|**m**|**s**}.

   For example, to set **the keep-alive interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config)> cloud drm keep_alive 600s
   (config)>
   ```

8. (Optional) Set the amount of time that the AnywhereUSB Plus device should wait between sending keep-alive messages to the Digi Remote Manager when using a cellular interface.

Allowed values are from 30 seconds to two hours. The default is 290 seconds.

```
(config)> cloud drm cellular_keep_alive value
(config)>
```

where *value* is any number of hours, minutes, or seconds, and takes the format ***number***{**h|m|s**}.

For example, to set **the cellular keep-alive interval** to ten minutes, enter either **10m** or **600s**:

```
(config)> cloud drm cellular_keep_alive 600s
(config)>
```

9. Set the number of allowed keep-alive misses. Allowed values are any integer between **2** and **64**. The default is **3**.

   ```
   (config)> cloud drm keep_alive_misses integer
   (config)>
   ```

10. The **watchdog** is used to monitor the connection to remote cloud services. If the connection is down, you can configure the device to restart the connection, or to reboot. The watchdog is enabled by default. To disable:

    ```
    (config)> cloud drm watchdog false
    (config)>
    ```

11. If **watchdog** is enabled:

    a. (Optional) Set the amount of time to wait before restarting the connection to the remote cloud services, once the connection is down.

       where *value* is any number of hours, minutes, or seconds, and takes the format ***number*** {**h|m|s**}.

       For example, to set **restart_timeout** to ten minutes, enter either **10m** or **600s**:

       ```
       (config)> cloud drm restart_timeout 600s
       (config)>
       ```

       The minimum value is 30 minutes and the maximum is 48 hours. If not set, this option is disabled. The default is 30 minutes.

    b. (Optional) Set the amount of time to wait before rebooting the device, once the connection to the remote cloud servicesis down. By default, this option is not set, which means that the option is disabled.

       where *value* is any number of hours, minutes, or seconds, and takes the format ***number*** {**h|m|s**}.

       For example, to set **reboot_timeout** to ten minutes, enter either **10m** or **600s**:

       ```
       (config)> cloud drm reboot_timeout 600s
       (config)>
       ```

       The minimum value is 30 minutes and the maximum is 48 hours. If not set, this option is disabled. The default is disabled.

12. (Optional) Determine whether to require a login and password to authenticate the user from the remote cloud services CLI:

```
(config)> cloud drm cli_local_auth true
(config)>
```

If set to **false**, no login prompt will be presented and the user will be logged in as **admin**. The default is **false**.

13. (Optional) Configure the AnywhereUSB Plus device to communicate with remote cloud services by using SMS:

    a. **Enable** SMS messaging:

    ```
    (config)> cloud drm sms enable true
    (config)>
    ```

    b. Set the phone number for Digi Remote Manager:

    ```
    (config)> cloud drm sms destination drm_phone_number
    (config)>
    ```

    c. (Optional) Set the service identifier:

    ```
    (config)> cloud drm sms sercice_id id
    (config)>
    ```

1. (Optional) Configure the AnywhereUSB Plus device to communicate with remote cloud services by using an HTTP proxy server:

    a. **Enable** the use of an HTTP proxy server:

    ```
    (config)> cloud drm proxy enable true
    (config)>
    ```

    b. Set the hostname of the proxy server:

    ```
    (config)> cloud drm proxy host hostname
    (config)>
    ```

    c. (Optional) Set the port number on the proxy server that the device should connect to. The default is 2138.

    ```
    (config)> cloud drm proxy port integer
    (config)>
    ```

14. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

15. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Collect device health data and set the sample interval

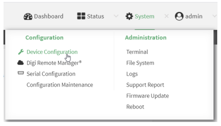
You can enable or disable the collection of device health data to upload to Digi Remote Manager, and configure the interval between health sample uploads. By default, device health data upload is enabled, and the health sample interval is set to 60 minutes.

To avoid a situation where several devices are uploading health metrics information to Remote Manager at the same time, the AnywhereUSB Plus device includes a preconfigured randomization of two minutes for uploading metrics. For example, if **Health sample interval** is set to five minutes, the metrics will be uploaded to Remote Manager at a random time between five and seven minutes.

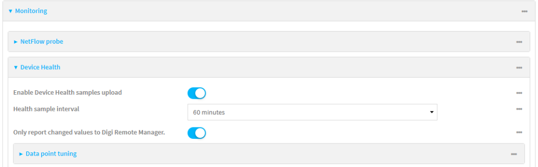
To disable the collection of device health data or enable it if it has been disabled, or to change the health sample interval:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Monitoring** > **Device Health**.

   Enable Device Health samples upload ●

   Health sample interval: 60 minutes

   Only report changed values to Digi Remote Manager. ●

   Device health data upload is enabled by default. To disable, click to toggle off **Enable Device Health samples upload**.
4. For **Health sample interval**, select the interval between health sample uploads.
5. **Only report changed values to Digi Remote Manager** is enabled by default.

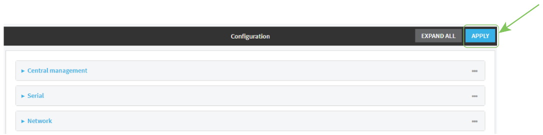
   When enabled:

   - The device only reports device health metrics that have changed health metrics were last uploaded. This is useful to reduce the bandwidth used to report health metrics.
   - All metrics are uploaded once every hour.

   When disabled, all metrics are uploaded every **Health sample interval**.
6. (Optional) Click to expand **Data point tuning**.

   Data point tuning options allow to you configure what data are uploaded to the Digi Remote Manager. All options are enabled by default.

7. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Device health data upload is enabled by default. To enable or disable:
   - To enable:

     ```
     (config)> monitoring devicehealth enable true
     (config)>
     ```
   - To disable:

     ```
     (config)> monitoring devicehealth enable false
     (config)>
     ```
4. The interval between health sample uploads is set to 60 minutes by default. To change:

   ```
   (config)> monitoring devicehealth interval value
   (config)>
   ```

   where *value* is one of **1**, **5**, **15**, **30**, or **60**, and represents the number of minutes between uploads of health sample data.
5. By default, the device will only report health metrics values to Digi Remote Manager that have changed health metrics were last uploaded. This is useful to reduce the bandwidth used to report health metrics. This is useful to reduce the bandwidth used to report health metrics. Even if enabled, all metrics are uploaded once every hour.

   To disable:

   ```
   (config)> monitoring devicehealth only_send_deltas false
   (config)>
   ```

   When disabled, all metrics are uploaded every **Health sample interval**.
6. (Optional) Tuning parameters allow to you configure what data are uploaded to the Digi Remote Manager. By default, all tuning parameters are enabled.

   To view a list of all available tuning parameters, use the **show** command:

```
(config)> show monitoring devicehealth tuning
all
       cellular
               rx
                       bytes
                               enable true
               tx
                       bytes
                               enable true
       eth
               rx
                       bytes
                               enable true
               tx
                       bytes
                               enable true
       serial
               rx
                       bytes
                               enable true
               tx
                       bytes
                               enable true
cellular
       1
               rx
                       bytes
                               enable true
                       packets
                               enable true
...
                                               (config)>
```

To disable a tuning parameter, set its value to false. For example, to turn off all reporting for the serial port:

```
(config)> monitoring devicehealth tuning all serial rx bytes enabled false
(config)> monitoring devicehealth tuning all serial tx bytes enabled false
(config)>
```

7. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Log into Digi Remote Manager

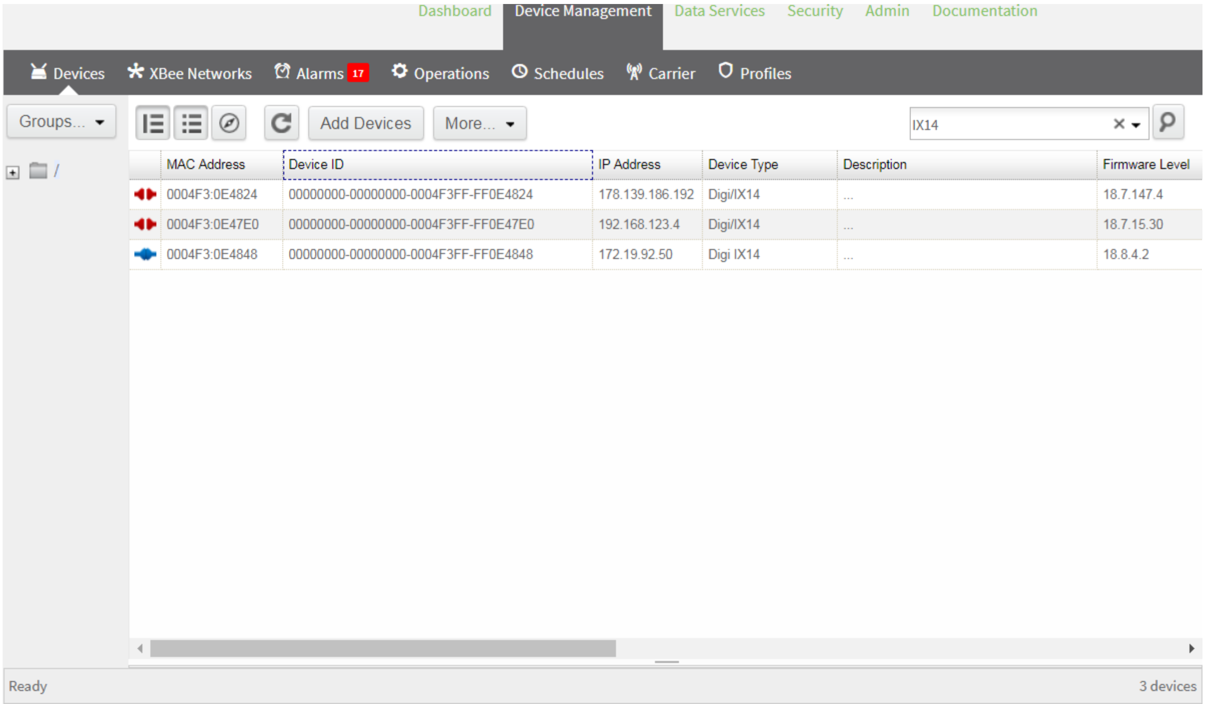
To start Digi Remote Manager

1. If you have not already done so, click here to sign up for a Digi Remote Manager account.
2. Check your email for Digi Remote Manager login instructions.
3. Go to remotemanager.digi.com.
4. Log into your Digi Remote Manager account.

Use Digi Remote Manager to view and manage your device

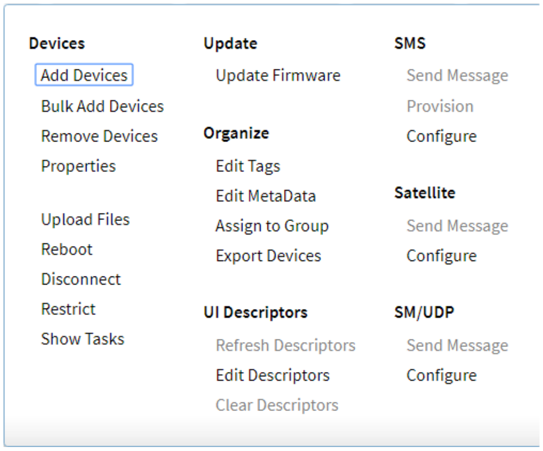
To view and manage your device:

1. If you have not already done so, connect to your Digi Remote Manager account.
2. Click **Device Management** to display a list of your devices.
3. Use the Search bar to locate the device you want to manage.



	MAC Address	Device ID	IP Address	Device Type	Description	Firmware Level
	0004F3:0E4824	00000000-00000000-0004F3FF-FF0E4824	178.139.186.192	Digi/IX14	...	18.7.147.4
	0004F3:0E47E0	00000000-00000000-0004F3FF-FF0E47E0	192.168.123.4	Digi/IX14	...	18.7.15.30
	0004F3:0E4848	00000000-00000000-0004F3FF-FF0E4848	172.19.92.50	Digi IX14	...	18.8.4.2

4. Select the device and click **Properties** to view general information for the device.
5. Click the **More** menu to perform a task.



Add a device to Digi Remote Manager

1. If you have not already done so, connect to your Digi Remote Manager account.
2. Click **Device Management** to display a list of your devices.
3. Click **Add Devices**.
4. Select **MAC Address** and enter the Ethernet MAC address for your device.
5. For **Install Code**, enter the default password on the printed label packaged with your device. The same default password is also shown on the label affixed to the bottom of the device.
6. Click **Add**.
7. Click **OK**.

Digi Remote Manager adds your AnywhereUSB Plus device to your account and it appears in the **Device Management** view.

View Digi Remote Manager connection status

To view the current Digi Remote Manager configuration:

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. The dashboard includes a Digi Remote Manager status pane:

⌨ Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. View the central management configuration:

```
(config)> show cloud
drm
        cellular_keep_alive 290s
        drm_url my.devicecloud.com
        keep_alive 60s
        keep_alive_misses 3
        retry_interval 30s
enable true
(config)>
```

1. Type **cancel** to exit configuration mode:

```
(config)> cancel
>
```

2. Type exit to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

To view the status of your device's connection to Remote Manager, use the show cloud command at the command line:

Command line

```
> show cloud

 Device Cloud Status
 -------------------

 Status    : Connected
 Server    : my.devicecloud.com
 Device ID : 00000000-00000000-0040FFFF-FF0F4594
>
```

The **Device ID** is the unique identifier for the device, as used by the Remote Manager.

Use the Digi Remote Manager mobile app

If you have a smart phone or tablet, you can use the Digi Remote Manager mobile app to automatically provision a new devices and monitor devices in your account.

To download the mobile app:

- For iPhone, go to the **App Store**
- For Android phones, go to **Google Play**

To sign up for a new Digi Remote Manager account using the mobile app:

1. From the menu, click **Log in or Sign Up**.
2. Click **Sign up** to create a new account.
3. You'll receive an email with login instructions.
4. From the **Digi Remote Manager** mobile app, click **Log in** and log into your new account.

To register a new device:

1. From the menu, select **Install a device with a QR or bar code** and scan the installation QR code on the label.
2. Follow the prompts to complete your AnywhereUSB Plus registration.

Digi Remote Manager registers your AnywhereUSB Plus and adds it to your Digi Remote Manager device list. You can now manage the device remotely using Digi Remote Manager.

Configure multiple devices using profiles

Digi recommends you take advantage of Digi Remote Manager profiles to manage multiple AnywhereUSB Plus routers. Typically, if you want to provision multiple AnywhereUSB Plus routers:

1. Using the AnywhereUSB Plus local WebUI, configure one AnywhereUSB Plus router to use as the model configuration for all subsequent AnywhereUSB Pluss you need to manage.
2. Register the configured AnywhereUSB Plus device in your Digi Remote Manager account.
3. In Digi Remote Manager, create a profile based on the configured AnywhereUSB Plus.
4. Apply the profile to the AnywhereUSB Plus devices you need to configure.

Digi Remote Manager provides multiple methods for applying profiles to registered devices. You can also include site-specific settings with a profile to override settings on a device-by-device basis.

Learn more

- For information on using Digi Remote Manager to configure and manage AnywhereUSB Plus routers, see the *Digi Remote Manager User Guide*.
- For information on using Digi Remote Manager APIs to develop custom applications, see the *Digi Remote Manager Programmer Guide*.

Diagnostics

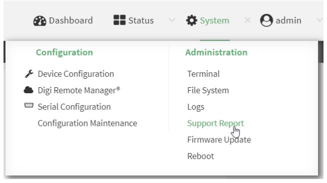
This chapter contains the following topics:

Generate a support report

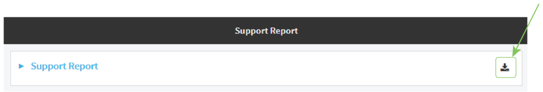
To generate and download a support report:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System**. Under **Administration**, click **Support Report**.



3. Click ⬇ to generate and download the support report.



Attach the support report to any support requests.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Use the **system support-report** command to generate the report:

```
> system support-report /var/log/
Saving support report to /var/log/support-report-0040D0133536-21-02-26-
8:04:23.bin
Support report saved.
>
```

3. Use the **scp** command to transfer the report to a remote host:

```
> scp host 192.168.4.1 user admin remote /home/admin/temp/ local
/var/log/support-report-00:40:D0:13:35:36-21-02-26-8:04:23.bin to remote
admin@192.168.4.1's password: adminpwd
support-report-0040D0133536-21-02-26-8:04:23.bin
>
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

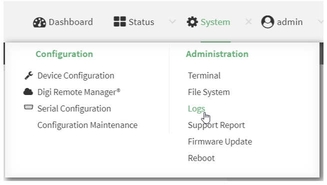
View system and event logs

See Configure options for the event and system logs for information about configuring the information displayed in event and system logs.

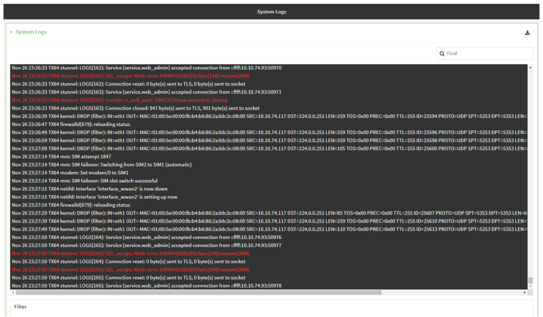
View System Logs

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System** > **Logs**.

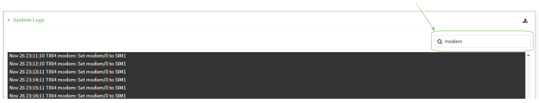


The system log displays:



```
Nov 26 23:26:33 TX64 stunnel: LOG5[162]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:50970
Nov 26 23:26:33 TX64 stunnel: LOG3[162]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:26:33 TX64 stunnel: LOG5[162]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:26:33 TX64 stunnel: LOG5[163]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:50971
Nov 26 23:26:33 TX64 stunnel: LOG3[163]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:26:33 TX64 stunnel: LOG5[163]: Connection closed: 947 byte(s) sent to TLS, 901 byte(s) sent to socket
Nov 26 23:26:39 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=259 TOS=0x00 PREC=0x00 TTL=255 ID=25594 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:26:47 TX64 firewalld[679]: reloading status
Nov 26 23:26:49 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=259 TOS=0x00 PREC=0x00 TTL=255 ID=25596 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:26:59 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=259 TOS=0x00 PREC=0x00 TTL=255 ID=25598 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:27:09 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=105 TOS=0x00 PREC=0x00 TTL=255 ID=25600 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:27:14 TX64 mm: SIM attempt 1847
Nov 26 23:27:14 TX64 mm: SIM failover: Switching from SIM2 to SIM1 (automatic)
Nov 26 23:27:14 TX64 modem: Set modem/0 to SIM1
Nov 26 23:27:14 TX64 mm: SIM failover: SIM slot switch successful
Nov 26 23:27:15 TX64 netifd: Interface 'interface_wwan2' is now down
Nov 26 23:27:15 TX64 netifd: Interface 'interface_wwan2' is setting up now
Nov 26 23:27:16 TX64 firewalld[679]: reloading status
Nov 26 23:27:29 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=85 TOS=0x00 PREC=0x00 TTL=255 ID=25607 PROTO=UDP SPT=5353 DPT=5353 LEN=6
Nov 26 23:27:39 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=259 TOS=0x00 PREC=0x00 TTL=255 ID=25610 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:27:49 TX64 kernel: DROP (filter): IN=eth1 OUT= MAC=01:00:5e:00:00:fb:b4:b6:86:2a:bb:3c:08:00 SRC=10.10.74.117 DST=224.0.0.251 LEN=110 TOS=0x00 PREC=0x00 TTL=255 ID=25613 PROTO=UDP SPT=5353 DPT=5353 LEN=
Nov 26 23:27:50 TX64 stunnel: LOG5[164]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:50976
Nov 26 23:27:50 TX64 stunnel: LOG5[165]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:50977
Nov 26 23:27:50 TX64 stunnel: LOG3[164]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:27:50 TX64 stunnel: LOG5[164]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:27:50 TX64 stunnel: LOG3[165]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:27:50 TX64 stunnel: LOG5[165]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:27:50 TX64 stunnel: LOG5[166]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:50978
```

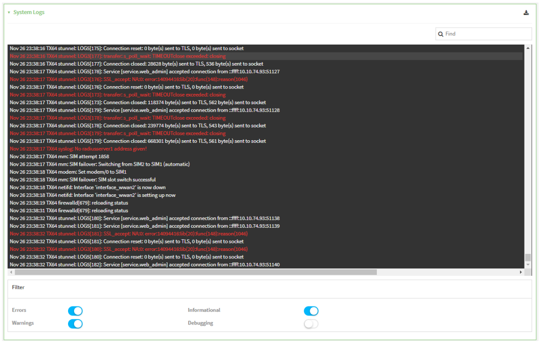
3. Limit the display in the system log by using the **Find** search tool.



modem

```
Nov 26 23:11:10 TX64 modem: Set modem/0 to SIM1
Nov 26 23:12:10 TX64 modem: Set modem/0 to SIM1
Nov 26 23:13:11 TX64 modem: Set modem/0 to SIM1
Nov 26 23:14:11 TX64 modem: Set modem/0 to SIM1
Nov 26 23:15:11 TX64 modem: Set modem/0 to SIM1
Nov 26 23:16:11 TX64 modem: Set modem/0 to SIM1
```

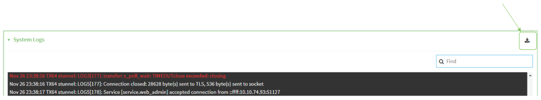
4. Use filters to configure the types of information displayed in the system logs.



```
Nov 26 23:38:16 TX64 stunnel: LOG5[175]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:38:16 TX64 stunnel: LOG3[177]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:38:16 TX64 stunnel: LOG5[177]: Connection closed: 28628 byte(s) sent to TLS, 536 byte(s) sent to socket
Nov 26 23:38:17 TX64 stunnel: LOG5[178]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51127
Nov 26 23:38:17 TX64 stunnel: LOG3[176]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:38:17 TX64 stunnel: LOG5[176]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:38:17 TX64 stunnel: LOG3[173]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:38:17 TX64 stunnel: LOG5[173]: Connection closed: 118374 byte(s) sent to TLS, 562 byte(s) sent to socket
Nov 26 23:38:17 TX64 stunnel: LOG5[179]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51128
Nov 26 23:38:17 TX64 stunnel: LOG3[178]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:38:17 TX64 stunnel: LOG5[178]: Connection closed: 239774 byte(s) sent to TLS, 543 byte(s) sent to socket
Nov 26 23:38:17 TX64 stunnel: LOG3[179]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:38:17 TX64 stunnel: LOG5[179]: Connection closed: 668301 byte(s) sent to TLS, 561 byte(s) sent to socket
Nov 26 23:38:17 TX64 syslog: No radiusserver1 address given!
Nov 26 23:38:17 TX64 mm: SIM attempt 1858
Nov 26 23:38:17 TX64 mm: SIM failover: Switching from SIM2 to SIM1 (automatic)
Nov 26 23:38:18 TX64 modem: Set modem/0 to SIM1
Nov 26 23:38:18 TX64 mm: SIM failover: SIM slot switch successful
Nov 26 23:38:18 TX64 netifd: Interface 'interface_wwan2' is now down
Nov 26 23:38:18 TX64 netifd: Interface 'interface_wwan2' is setting up now
Nov 26 23:38:19 TX64 firewalld[679]: reloading status
Nov 26 23:38:31 TX64 firewalld[679]: reloading status
Nov 26 23:38:32 TX64 stunnel: LOG5[180]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51138
Nov 26 23:38:32 TX64 stunnel: LOG5[181]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51139
Nov 26 23:38:32 TX64 stunnel: LOG3[181]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:38:32 TX64 stunnel: LOG5[181]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:38:32 TX64 stunnel: LOG3[180]: SSL_accept: NA:0: error:14094416:lib(20):func(148):reason(1046)
Nov 26 23:38:32 TX64 stunnel: LOG5[180]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
Nov 26 23:38:32 TX64 stunnel: LOG5[182]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51140
```

Errors ● Informational ●
Warnings ● Debugging ○

5. Click ⤓ to download the system log.



```
Nov 26 23:38:16 TX64 stunnel: LOG3[177]: transfer: s_poll_wait: TIMEOUTclose exceeded: closing
Nov 26 23:38:16 TX64 stunnel: LOG5[177]: Connection closed: 28628 byte(s) sent to TLS, 536 byte(s) sent to socket
Nov 26 23:38:17 TX64 stunnel: LOG5[178]: Service [service.web_admin] accepted connection from ::ffff:10.10.74.93:51127
```

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Use show log at the Admin CLI prompt:

```
> show log

Timestamp        Message
--------------  ------------------------------------------------------------
--
Nov 26 21:54:34  AnywhereUSB Plus netifd: Interface 'interface_wan' is
setting up now
Nov 26 21:54:35  AnywhereUSB Plus firewalld[621]: reloading status
...
>
```

3. (Optional) Use the **show log number *num*** command to limit the number of lines that are displayed. For example, to limit the log to the most recent ten lines:

```
> show log number 10

Timestamp        Message
--------------  ------------------------------------------------------------
--
Nov 26 21:54:34  AnywhereUSB Plus netifd: Interface 'interface_wan' is
setting up now
Nov 26 21:54:35  AnywhereUSB Plus firewalld[621]: reloading status
...
>
```

4. (Optional) Use the **show log filter *value*** command to limit the number of lines that are displayed. Allowed values are **critical**, **warning**, **info**, and **debug**. For example, to limit the event list to only info messages:

```
> show log filter info

Timestamp        Type     Category   Message
---------------- -------  ---------  ----------------------------------------
---
Nov 26 22:01:26  info     user
name=admin~service=cli~state=opened~remote=192.168.1.2
Nov 26 22:01:25  info     user
name=admin~service=cli~state=closed~remote=192.168.1.2
```

```
...
>
```

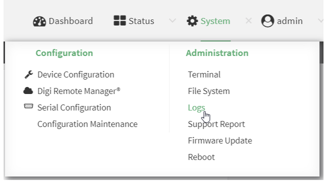
5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

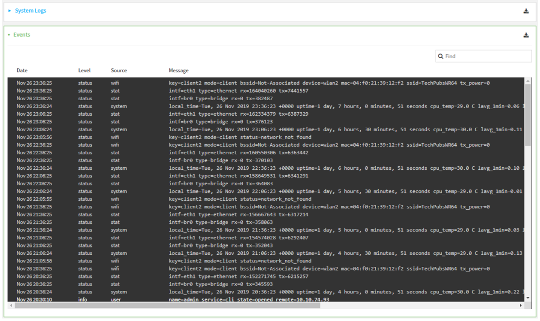
View Event Logs

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the main menu, click **System** > **Logs**.

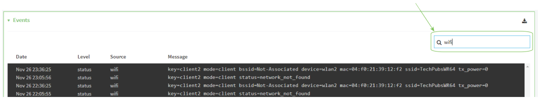


3. Click ▾ **System Logs** to collapse the system logs viewer, or scroll down to **Events**.
4. Click ▸ **Events** to expand the event viewer.



```
Date            Level   Source  Message
Nov 26 23:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 23:36:25 status  stat    intf=eth1 type=ethernet rx=164040260 tx=7441557
Nov 26 23:36:25 status  stat    intf=br0 type=bridge rx=0 tx=382487
Nov 26 23:36:24 status  system  local_time=Tue, 26 Nov 2019 23:36:23 +0000 uptime=1 day, 7 hours, 0 minutes, 51 seconds cpu_temp=29.0 C lavg_1min=0.06 l
Nov 26 23:06:25 status  stat    intf=eth1 type=ethernet rx=162334379 tx=6387329
Nov 26 23:06:25 status  stat    intf=br0 type=bridge rx=0 tx=376123
Nov 26 23:06:24 status  system  local_time=Tue, 26 Nov 2019 23:06:23 +0000 uptime=1 day, 6 hours, 30 minutes, 51 seconds cpu_temp=30.0 C lavg_1min=0.11
Nov 26 23:05:56 status  wifi    key=client2 mode=client status=network_not_found
Nov 26 22:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 22:36:25 status  stat    intf=eth1 type=ethernet rx=160550306 tx=6363442
Nov 26 22:36:25 status  stat    intf=br0 type=bridge rx=0 tx=370103
Nov 26 22:36:24 status  system  local_time=Tue, 26 Nov 2019 22:36:23 +0000 uptime=1 day, 6 hours, 0 minutes, 51 seconds cpu_temp=30.0 C lavg_1min=0.10 l
Nov 26 22:06:25 status  stat    intf=eth1 type=ethernet rx=158649531 tx=6341291
Nov 26 22:06:25 status  stat    intf=br0 type=bridge rx=0 tx=364083
Nov 26 22:06:24 status  system  local_time=Tue, 26 Nov 2019 22:06:23 +0000 uptime=1 day, 5 hours, 30 minutes, 51 seconds cpu_temp=29.0 C lavg_1min=0.01
Nov 26 22:05:55 status  wifi    key=client2 mode=client status=network_not_found
Nov 26 21:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 21:36:25 status  stat    intf=eth1 type=ethernet rx=156667643 tx=6317214
Nov 26 21:36:25 status  stat    intf=br0 type=bridge rx=0 tx=358063
Nov 26 21:36:24 status  system  local_time=Tue, 26 Nov 2019 21:36:23 +0000 uptime=1 day, 5 hours, 0 minutes, 51 seconds cpu_temp=29.0 C lavg_1min=0.03 l
Nov 26 21:06:25 status  stat    intf=eth1 type=ethernet rx=154574028 tx=6292407
Nov 26 21:06:25 status  stat    intf=br0 type=bridge rx=0 tx=352043
Nov 26 21:06:24 status  system  local_time=Tue, 26 Nov 2019 21:06:23 +0000 uptime=1 day, 4 hours, 30 minutes, 51 seconds cpu_temp=29.0 C lavg_1min=0.13
Nov 26 21:05:58 status  wifi    key=client2 mode=client status=network_not_found
Nov 26 20:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 20:36:25 status  stat    intf=eth1 type=ethernet rx=152271745 tx=6215257
Nov 26 20:36:25 status  stat    intf=br0 type=bridge rx=0 tx=345593
Nov 26 20:36:24 status  system  local_time=Tue, 26 Nov 2019 20:36:23 +0000 uptime=1 day, 4 hours, 0 minutes, 51 seconds cpu_temp=30.0 C lavg_1min=0.22 l
Nov 26 20:30:10 info    user    name=admin service=cli state=opened remote=10.10.74.93
```

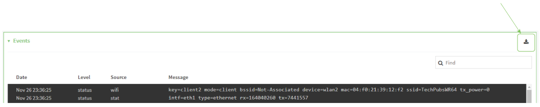
5. Limit the display in the event log by using the **Find** search tool.



Find: wifi

```
Date            Level   Source  Message
Nov 26 23:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 23:05:56 status  wifi    key=client2 mode=client status=network_not_found
Nov 26 22:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 22:05:55 status  wifi    key=client2 mode=client status=network_not_found
```

6. Click 📥 to download the event log.



```
Date            Level   Source  Message
Nov 26 23:36:25 status  wifi    key=client2 mode=client bssid=Not-Associated device=wlan2 mac=04:f0:21:39:12:f2 ssid=TechPubsWR64 tx_power=0
Nov 26 23:36:25 status  stat    intf=eth1 type=ethernet rx=164040260 tx=7441557
```

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Use show event at the Admin CLI prompt:

```
> show event

Timestamp        Type    Category  Message
---------------- ------- --------- ----------------------------------------
---
Nov 26 21:42:37  status  stat
intf=eth1~type=ethernet~rx=11332435~tx=5038762
Nov 26 21:42:35  status  system    local_time=Thu, 08 Aug 2019 21:42:35
+0000~uptime=3 hours, 0 minutes, 48 seconds
...
>
```

3. (Optional) Use the **show event number *num*** command to limit the number of lines that are displayed. For example, to limit the event list to the most recent ten lines:

```
> show event number 10

Timestamp        Type    Category  Message
---------------- ------- --------- ----------------------------------------
---
Nov 26 21:42:37  status  stat
intf=eth1~type=ethernet~rx=11332435~tx=5038762
Nov 26 21:42:35  status  system    local_time=Thu, 08 Aug 2019 21:42:35
+0000~uptime=3 hours, 0 minutes, 48 seconds
...
>
```

4. (Optional) Use the **show event table *value*** command to limit the number of lines that are displayed. Allowed values are **error**, **info**, and **status**. For example, to limit the event list to only info messages:

```
> show event table info

Timestamp        Type    Category  Message
---------------- ------- --------- ----------------------------------------
---
Nov 26 22:01:26  info    user
name=admin~service=cli~state=opened~remote=192.168.1.2
Nov 26 22:01:25  info    user
name=admin~service=cli~state=closed~remote=192.168.1.2
...
>
```

5. Type **exit** to exit the Admin CLI.

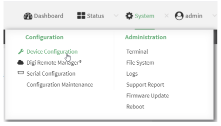
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure syslog servers

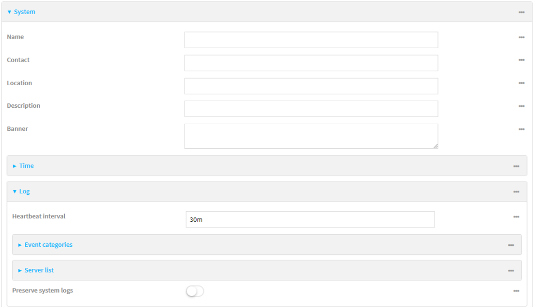
You can configure remote syslog servers for storing event and system logs.

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

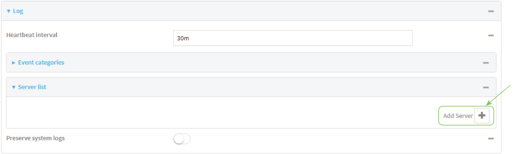
   

   The **Configuration** window is displayed.
3. Click **System** > **Log**.

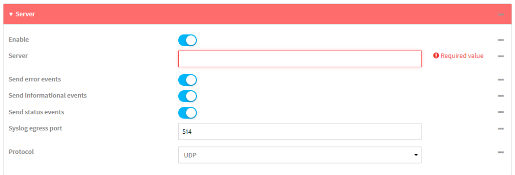
   Heartbeat interval 30m

4. Add and configure a remote syslog server:
   a. Click to expand **Server list**.
   b. For **Add Server**, click ✚.

   Heartbeat interval 30m

   The log server configuration window is displayed.

   Syslog egress port 514

   Protocol UDP

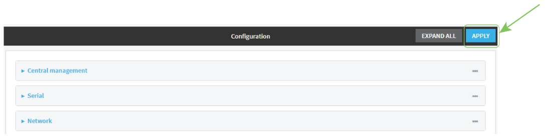
   Log servers are enabled by default. To disable, click to toggle off **Enable**.
   c. Type the host name or IP address of the **Server**.

d. Select the event categories that will be sent to the server. By default, all event categories are enabled. You can disable logging for error, informational, and status event categories by clicking to toggle off the category.

e. For **Syslog egress port**, type the port number to use for the syslog server. The default is **514**.

f. For **Protocol**, select the IP protocol to use for communication with the syslog server. Available options are **TCP** and **UPD**. The default is **UPD**.

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. (Optional) To configure remote syslog servers:

   a. Add a remote server:

```
(config)> add system log remote end
(config system log remote 0)>
```

   b. Enable the server:

```
(config system log remote 0)> enable true
(config system log remote 0)>
```

   c. Set the host name or IP address of the server:

```
(config system log remote 0)> server hostname
(config system log remote 0)>
```

   d. The event categories that will be sent to the server are automatically enabled when the server is enabled.

   - To disable informational event messages:

```
(config system log remote 0)> info false
(config system log remote 0)>
```

- To disable status event messages:

  ```
  (config system log remote 0)> status false
  (config system log remote 0)>
  ```

- To disable informational event messages:

  ```
  (config system log remote 0)> error false
  (config system log remote 0)>
  ```

4. Set the port number to use for the syslog server:

   ```
   (config system log remote 0)> port value
   (config system log remote 0)>
   ```

   where *value* is any integer between **1** and **65535**. The default is **514**.
5. Set the IP protocol to use for communication with the syslog server:

   ```
   (config system log remote 0)> protocol value
   (config system log remote 0)>
   ```

   where *value* is either **tcp** or **udp**. The default is **udp**.
6. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure options for the event and system logs

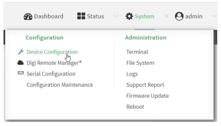
The default configuration for event and system logging is:

- The heartbeat interval, which determines the amount of time to wait before sending a heartbeat event if no other events have been sent, is set to 30 minutes.
- All event categories are enabled.

To change or disable the heartbeat interval, or to disable event categories, and to perform other log configuration:

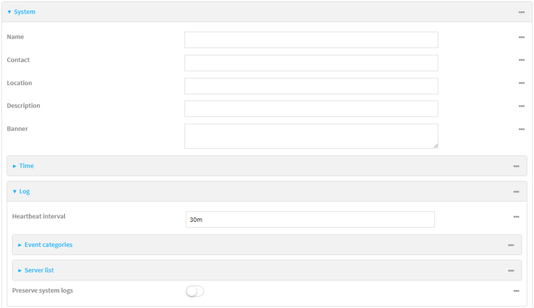
☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.

3. Click **System** > **Log**.



Heartbeat interval 30m

4. (Optional) To change the **Heartbeat interval** from the default of 30 minutes, type a new value. The heartbeat interval determines the amount of time to wait before sending a heartbeat event if no other events have been sent.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **Heartbeat interval** to ten minutes, enter **10m** or **600s**.

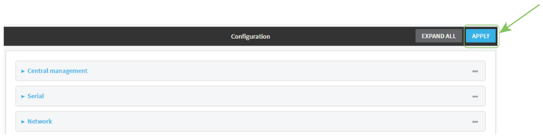
   To disable the **Heartbeat interval**, enter **0s**.

5. (Optional) To disable event categories, or to enable them if they have been disabled:
   a. Click to expand **Event Categories**.
   b. Click an event category to expand.
   c. Depending on the event category, you can enable or disable informational events, status events, and error events. Some categories also allow you to set the **Status interval**, which is the time interval between periodic status events.
6. (Optional) See Configure syslog servers for information about configuring remote syslog servers to which log messages will be sent.
7. Enable **Preserve system logs** to save the current session's system log after a reboot.

   By default, the AnywhereUSB Plus device erases system logs each time the device is powered off or rebooted.

   Note You should only enable **Preserve system logs** temporarily to debug issues. Once you are finished debugging, immediately disable **Preserve system logs** to avoid unnecessary wear to the flash memory.

8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. (Optional) To change the heartbeat interval from the default of 30 minutes, set a new value. The heartbeat interval determines the amount of time to wait before sending a heartbeat event if no other events have been sent.

   ```
   (config)> system log heartbeat_interval value
   (config)>
   ```

   where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **the heartbeat interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config)> system log heartbeat_interval 600s
   (config)>
   ```

   To disable the heartbeat interval, set the value to **0s**

4. Enable preserve system logs functionality to save the current session's system log after a reboot. By default, the AnywhereUSB Plus device erases system logs each time the device is powered off or rebooted.

   Note You should only enable **Preserve system logs** temporarily to debug issues. Once you are finished debugging, immediately disable **Preserve system logs** to avoid unnecessary wear to the flash memory.

   ```
   (config)> system log persistent true
   (config)>
   ```

5. (Optional) To disable event categories, or to enable them if they have been disabled:

   a. Use the question mark (**?**) to determine available event categories:

      ```
      (config)> system log event ?

      Event categories: Settings to enable individual event categories.

       Additional Configuration
      ```

```
 ---------------------------------------------------------------------
--------
 arping                    ARP ping
 config                    Configuration
 dhcpserver                DHCP server
 firmware                  Firmware
 location                  Location
 modem                     Modem
 netmon                    Active recovery
 network                   Network interfaces
 openvpn                   OpenVPN
 portal                    Captive portal
 remote                    Remote control
 restart                   Restart
 serial                    Serial
 sms                       SMS commands
 speed                     Speed
 stat                      Network statistics
 user                      User
 wol                       Wake-On-LAN

(config)> system log event
```

b. Depending on the event category, you can enable or disable informational events, status events, and error events. Some categories also allow you to set the status interval, which is the time interval between periodic status events. For example, to configure DHCP server logging:

   i. Use the question mark (**?**) to determine what events are available for DHCP server logging configuration:

```
(config)> system log event dhcpserver ?
...
DHCP server: Settings for DHCP server events. Informational events
are generated
when a lease is obtained or released. Status events report the
current list of
leases.

 Parameters                Current Value
 ---------------------------------------------------------------------
------------
 info                      true          Enable informational events
 status                    true          Enable status events
 status_interval           30m           Status interval

(config)> system log event dhcpserver
```

ii. To disable informational messages for the DHCP server:

```
(config)> system log event dhcpserver info false
(config)>
```

iii. To change the status interval:

```
(config)> system log event dhcpserver status_interval value
(config)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **the status interval** to ten minutes, enter either **10m** or **600s**:

```
(config)> system log event dhcpserver status_interval 600s
(config)>
```

6. (Optional) See Configure syslog servers for information about configuring remote syslog servers to which log messages will be sent.
7. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Analyze network traffic

The AnywhereUSB Plus device includes a network analyzer tool that captures data traffic on any interface and decodes the captured data traffic for diagnostics. You can capture data traffic on multiple interfaces at the same time and define capture filters to reduce the captured data. You can capture up to 10 MB of data traffic in two 5 MB files per interface.

To perform a more detailed analysis, you can download the captured data traffic from the device and view it using a third-party application.

Note Data traffic is captured to RAM and the captured data is lost when the device reboots unless you save the data to a file. See Save captured data traffic to a file.

This section contains the following topics:

Configure packet capture for the network analyzer

To use the network analyzer, you must create one or more packet capture configuration.

Required configuration items

- The interface used by this packet capture configuration.

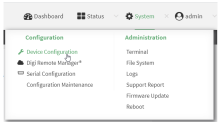
Additional configuration items

- The filter expression for this packet capture configuration.
- Schedule the analyzer to run based on a specified event or at a particular time:
  - The events or time that will trigger the analyzer to run, using this capture configuration.
  - The amount of time that the analyzer session will run.
  - The frequency with which captured events will be saved.

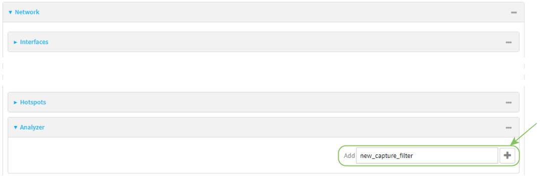
To configure a packet capture configuration:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

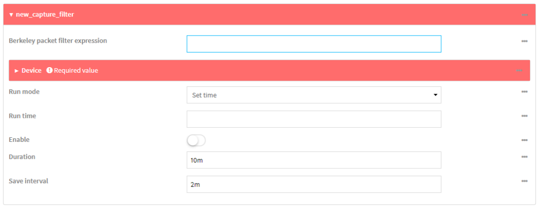
   

   The **Configuration** window is displayed.
3. Click **Network** > **Analyzer**.
4. For **Add Capture settings**, type a name for the capture filter and click ✚.

   Add new_capture_filter

   The new capture filter configuration is displayed.

   ▾ new_capture_filter

   Berkeley packet filter expression

   ▸ Device ❗ Required value

   Run mode: Set time

   Run time

   Enable

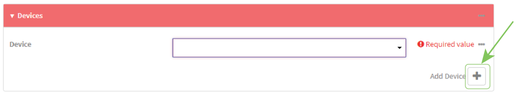
   Duration: 10m

   Save interval: 2m

5. Add one or more interface to the capture filter:

    a. Click to expand **Device**.

    b. Click ✚ to add an interface to the capture setting instance.

    c. For **Device**, select an interface.

    d. Repeat to add additional interfaces to the capture filter.

6. (Optional) For **Berkeley packet filter expression**, type a filter using Berkeley Packet Filter (BPF) syntax. See Example filters for capturing data traffic for examples of filters using BPF syntax.

7. (Optional) Schedule the analyzer to run, using this capture filter, based on a specified event or at a particular time:

    a. For **Run mode**, select the mode that will be used to run the capture filter. Available options are:

    - **On boot**: The capture filter will run once each time the device boots.
    - **Interval**: The capture filter will start running at the specified interval, within 30 seconds after the configuration change is saved.
        - If **Interval** is selected, in **Interval**, type the interval.

          Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

          For example, to set **Interval** to ten minutes, enter **10m** or **600s**.
    - **Set time**: Runs the capture filter at a specified time of the day.
        - If **Set Time** is selected, specify the time that the capture filter should run in **Run time**, using the format *HH:MM*.
    - **During system maintenance**: The capture filter will run during the system maintenance time window.

    b. **Enable** the capture filter schedule.

    c. For **Duration**, type the amount of time that the scheduled analyzer session will run.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

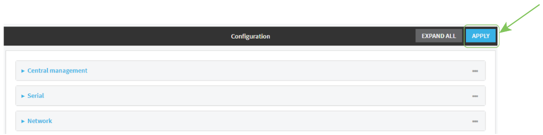
    For example, to set **Duration** to ten minutes, enter **10m** or **600s**.

    d. For **Save interval**, type the frequency with which captured events will be saved.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **Save interval** to ten minutes, enter **10m** or **600s**.

8. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Add a new capture filter:

   ```
   (config)> add network analyzer name
   (config network analyzer name)>
   ```

4. Add an interface to the capture filter:

   ```
   (config network analyzer name)> add device end device
   (config network analyzer name)>
   ```

   Determine available devices and the proper syntax.

   To determine available devices and proper syntax, use the space bar autocomplete feature:

   ```
   (config network analyzer name)> add device end <space>
   /network/device/eth1                /network/device/loopback
   /network/interface/defaultip        /network/interface/defaultlinklocal
   /network/interface/eth1             /network/interface/loopback
   /network/interface/modem
   (config network analyzer name)> add interface end /network/
   ```

   Repeat to add additional interfaces.

5. (Optional) Set a filter for the capture filter:

   ```
   (config network analyzer name)> filter value
   (config network analyzer name)>
   ```

   where *value* is a filter using Berkeley Packet Filter (BPF) syntax. Values that contain spaces must be enclosed in double quotes (**"**).

   See Example filters for capturing data traffic for examples of filters using BPF syntax.

6. (Optional) Schedule the analyzer to run, using this capture filter, based on a specified event or at a particular time:

   a. Enable scheduling for this capture filter:

   ```
   (config network analyzer name)> schedule enable true
   (config network analyzer name)>
   ```

   b. Set the mode that will be used to run the capture filter:

   ```
   (config network analyzer name)> when mode
   (config network analyzer name)>
   ```

where *mode* is one of the following:

- **boot**: The script will run once each time the device boots.
- **interval**: The script will start running at the specified interval, within 30 seconds after the configuration change is saved. If **interval** is selected, set the interval:

  ```
  (config add network analyzer name)> on_interval value
  (config add network analyzer name)>
  ```

  where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

  For example, to set **on_interval** to ten minutes, enter either **10m** or **600s**:

  ```
  (config network analyzer name)> on_interval  600s
  (config network analyzer name)>
  ```

- **set_time**: Runs the script at a specified time of the day. If **set_time** is set, set the time that the script should run, using the format *HH*:*MM*:

  ```
  (config network analyzer name)> run_time HH:MM
  (config network analyzer name)>
  ```

- **maintenance_time**: The script will run during the system maintenance time window.

c. Set the amount of time that the scheduled analyzer session will run:

```
(config network analyzer name)> duration value
(config network analyzer name)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **duration** to ten minutes, enter either **10m** or **600s**:

```
(config network analyzer name)> save_interval 600s
(config network analyzer name)>
```

d. Set the frequency with which captured events will be saved:

```
(config network analyzer name)> save_interval value
(config network analyzer name)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **save_interval** to ten minutes, enter either **10m** or **600s**:

```
(config network analyzer name)> save_interval 600s
(config network analyzer name)>
```

7. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Example filters for capturing data traffic

The following are examples of filters using Berkeley Packet Filter (BPF) syntax for capturing several types of network data. See https://biot.com/capstats/bpf.html for detailed information about BPF syntax.

Example IPv4 capture filters

- Capture traffic to and from IP host 192.168.1.1:

  ```
  ip host 192.168.1.1
  ```

- Capture traffic from IP host 192.168.1.1:

  ```
  ip src host 192.168.1.1
  ```

- Capture traffic to IP host 192.168.1.1:

  ```
  ip dst host 192.168.1.1
  ```

- Capture traffic for a particular IP protocol:

  ```
  ip proto protocol
  ```

  where *protocol* is a number in the range of **1** to **255** or one of the following keywords: **icmp**, **icmp6**, **igmp**, **pim**, **ah**, **esp**, **vrrp**, **udp**, or **tcp**.

- Capture traffic to and from a TCP port 80:

  ```
  ip proto tcp and port 80
  ```

- Capture traffic to UDP port 53:

  ```
  ip proto udp and dst port 53
  ```

- Capture traffic from UDP port 53:

  ```
  ip proto udp and src port 53
  ```

- Capture to and from IP host 10.0.0.1 but filter out ports 22 and 80:

  ```
  ip host 10.0.0.1 and not (port 22 or port 80)
  ```

Example Ethernet capture filters

- Capture Ethernet packets to and from a host with a MAC address of 00:40:D0:13:35:36:

```
ether host 00:40:D0:13:35:36
```

- Capture Ethernet packets from host 00:40:D0:13:35:36:

```
ether src 00:40:D0:13:35:36:
```

- Capture Ethernet packets to host 00:40:D0:13:35:36:

```
ether dst 00:40:D0:13:35:36
```

Capture packets from the command line

You can start packet capture at the command line with the analyzer start command. Alternatively, you can schedule the network analyzer to run based on a specified event or at a particular time. See Configure packet capture for the network analyzer for information about scheduling packet capturing.

Additional analyzer commands allow you to:

- Stop capturing packets.
- Save captured data traffic to a file.
- Clear captured data.

Required configuration items

- A configured packet capture. See Configure packet capture for the network analyzer for packet capture configuration information.

To start packet capture from the command line:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Type the following at the Admin CLI prompt:

   ```
   > analyzer start name capture_filter
   >
   ```

   where *capture_filter* is the name of a packet capture configuration. See Configure packet capture for the network analyzer for more information.

   To determine available packet capture configurations, use the **?**:

   ```
   > analyzer start name ?

   name: Name of the capture filter to use.
   Format:
     test_capture
   ```

```
  capture_ping

> analyzer start name
```

You can capture up to 10 MB of data traffic in two 5 MB files per interface.

Note Data traffic is captured to RAM and the captured data is lost when the device reboots unless you save the data to a file. See Save captured data traffic to a file.

Stop capturing packets

You can stop packet capture at the command line with the analyzer stop command.

To stop packet capture from the command line:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Type the following at the Admin CLI prompt:

   ```
   > analyzer stop name capture_filter
   >
   ```

   where *capture_filter* is the name of a packet capture configuration. See Configure packet capture for the network analyzer for more information.

   To determine available packet capture configurations, use the **?**:

   ```
   > analyzer stop name ?

   name: Name of the capture filter to use.
   Format:
     test_capture
     capture_ping

   > analyzer stop name
   ```

Show captured traffic data

To view captured data traffic, use the show analyzer command. The command output show the following information for each packet:

- The packet number.
- The timestamp for when the packet was captured.
- The length of the packet and the amount of data captured.
- Whether the packet was sent or received by the device.
- The interface on which the packet was sent or received.

- A hexadecimal dump of the packet of up to 256 bytes.
- Decoded information of the packet.

To show captured data traffic:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. Type the following at the Admin CLI prompt:

```
> show analyzer name capture_filter

Packet 1 : Feb-26-2021 8:04:23.287682, Length 60 bytes (Captured Length 60
bytes)

Received on interface eth1

      00 40 ff 80 01 20 b4 b6  86 21 b5 73 08 00 45 00   .@... .. .!.s..E.
      00 28 3d 36 40 00 80 06  14 bc 0a 0a 4a 82 0a 0a   .(=6@... ....J..
      4a 48 cd ae 00 16 a4 4b  ff 5f ee 1f d8 23 50 10   JH.....K ._...#P.
      08 02 c7 40 00 00 00 00  00 00 00 00               ...@.... ....

    Ethernet Header
      Destination MAC Addr : 00:40:D0:13:35:36
      Source MAC Addr      : fb:03:53:05:11:2f
      Ethernet Type        : IP (0x0800)
    IP Header
      IP Version           : 4
      Header Length        : 20 bytes
      ToS                  : 0x00
      Total Length         : 40 bytes
      ID                   : 15670 (0x3d36)
      Flags                : Do not fragment
      Fragment Offset      : 0 (0x0000)
      TTL                  : 128 (0x80)
      Protocol             : TCP (6)
      Checksum             : 0x14bc
      Source IP Address    : 10.10.74.130
      Dest. IP Address     : 10.10.74.72
    TCP Header
      Source Port          : 52654
      Destination Port     : 22
      Sequence Number      : 2756443999
      Ack Number           : 3995064355
      Data Offset          : 5
      Flags                : ACK
      Window               : 2050
      Checksum             : 0xc740
      Urgent Pointer       : 0
    TCP Data
      00 00 00 00 00 00                                           ......

>
```

where *capture_filter* is the name of a packet capture configuration. See Configure packet capture for the network analyzer for more information.

To determine available packet capture configurations, use the **?**:

```
> show anaylzer name ?

name: Name of the capture filter to use.
Format:
  test_capture
  capture_ping

> show anaylzer name
```

Save captured data traffic to a file

Data traffic is captured to RAM and when the device reboots, the data is lost. To retain the captured data, first save the data to a file and then upload the file to a PC.

To save captured traffic data to a file, use the analyzer save command:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Type the following at the Admin CLI prompt:

   ```
   > analyzer save filename filename name capture_filter
   >
   ```

   where:

   - *filename* is the name of the file that the captured data will be saved to.

     Determine filenames already in use:

     Use the tab autocomplete feature to determine filenames that are currently in use:

     ```
     > analyzer save name <tab>
     test1_analyzer_capture     test2_analyzer_capture
     > analyzer save name
     ```

   - *capture_filter* is the name of a packet capture configuration. See Configure packet capture for the network analyzer for more information.

     To determine available packet capture configurations, use the **?**:

     ```
     > analyzer save name ?
     ```

```
name: Name of the capture filter to use.
Format:
  test_capture
  capture_ping

> analyzer save name
```

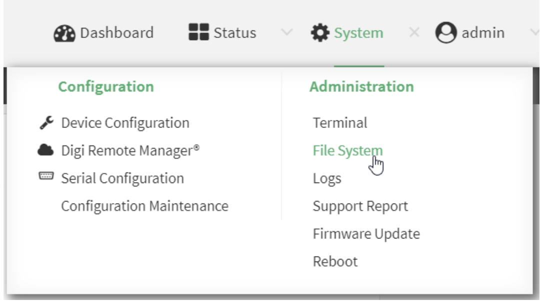
The file is stored in the **/etc/config/analyzer** directory. To transfer the file to your PC, see Download captured data to your PC.

Download captured data to your PC

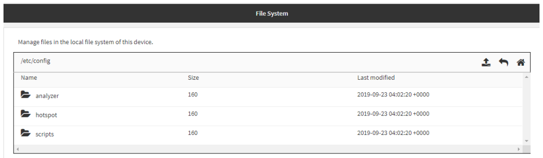
After saving captured data to a file (see Save captured data traffic to a file), you can download the file from the WebUI or from the command line by using the scp (secure copy file) command.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **System**. Under **Administration**, click **File System**.



The **File System** page appears.



File System

Manage files in the local file system of this device.

/etc/config

Name	Size	Last modified
analyzer	160	2019-09-23 04:02:20 +0000
hotspot	160	2019-09-23 04:02:20 +0000
scripts	160	2019-09-23 04:02:20 +0000

3. Highlight the **analyzer** directory and click ➦ to open the directory.
4. Select the saved analyzer report you want to download and click ⤓ (download).

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Type **scp** to use the Secure Copy program to copy the file to your PC:

```
> scp host hostname-or-ip user username remote remote-path local local-path
to remote
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the location on the remote host where the file will be copied.
- *local-path* is the path and filename on the AnywhereUSB Plus device.

To download the traffic saved in the file **/etc/config/analyzer/eth0.pcpng** to a PC with the IP **192.168.210.2**, for a user named **maria**, to the **/home/maria** directory:

```
> scp host 192.168.210.2 user maria remote /home/maria local
/etc/config/analyzer/eth0.pcpng to remote

maria@192.168.210.2's password:
eth0.pcpng                                   100%   11KB 851.3KB/s   00:00
```

Clear captured data

To clear captured data traffic in RAM, use the analyzer clear command:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. Type the following at the Admin CLI prompt:

```
> analyzer clear name capture_filter
>
```

where *capture_filter* is the name of a packet capture configuration. See Configure packet capture for the network analyzer for more information.

To determine available packet capture configurations, use the **?**:

```
> anaylzer clear name ?

name: Name of the capture filter to use.
Format:
  test_capture
  capture_ping

> anaylzer clear name
```

Note You can remove data traffic saved to a file using the rm command.

Use the ping command to troubleshoot network connections

Use the ping command troubleshoot connectivity problems.

Ping to check internet connection

To check your internet connection:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type the ping command followed by the host name or IP address of the server to be pinged:

```
> ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=54 time=11.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=54 time=10.7 ms
...
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Stop ping commands

To stop pings when the number of pings to send (the **count** parameter) has been set to a high value, enter **Ctrl+C**.

Use the traceroute command to diagnose IP routing problems

Use the **traceroute** command to diagnose IP routing problems. This command traces the route to a remote IP host and displays results. The **traceroute** command differs from ping in that traceroute shows where the route fails, while ping simply returns a single error on failure.

See the traceroute command description for command syntax and examples. The **traceroute** command has several parameters. Only **host** is required.

- **host**: The IP address of the destination host.
- **bypass**: Send directly to a host on an attached network.
- **debug**: Enable socket level debugging.
- **dontfragment**: Do not fragment probe packets.
- **first_ttl**: Specifies with what TTL to start. (Default: 1)
- **gateway**: Route the packet through a specified gateway.
- **icmp**: Use ICMP ECHO for probes.
- **interface**: Specifies the interface.

- **ipchecksums**: Calculate ip checksums.
- **max_ttl**: Specifies the maximum number of hops. (Default: 30)
- **nomap**: Do not map IP addresses to host names
- **nqueries**: Sets the number of probe packets per hop. (Default: 3)
- **packetlen**: Total size of the probing packet. (Default: -1)
- **pausemsecs**: Minimal time interval between probes (Default: 0)
- **port**: Specifies the destination port. (Default: -1)
- **src_addr**: Chooses an alternative source address.
- **tos**: Set Type of Service. (Default: -1)
- **verbose**: Verbose output.
- **waittime**: Max wait for a response to a probe. (Default: 5)

Example

This example shows using **traceroute** to verify that the AnywhereUSB device can route to host **8.8.8.8** (www.google.com) through the default gateway. The command output shows that **15** routing hops were required to reach the host:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, use the **traceroute** command to view IP routing information:

```
> traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 52 byte packets
 1  192.168.8.1 (192.168.8.1)  0 ms  0 ms  0 ms
 2  10.10.10.10 (10.10.10.10)  0 ms  2 ms  2 ms
 3  * 10.10.8.23 (10.10.8.23)  1 ms  1 ms
 4  96.34.84.22 (96.34.84.22)  1 ms  1 ms  1 ms
 5  96.34.81.190 (96.34.81.190)  2 ms  2 ms  2 ms
 6  * * *
 7  96.34.2.12 (96.34.2.12)  11 ms  11 ms  11 ms
 8  * * *
 9  8.8.8.8 (8.8.8.8)  11 ms  11 ms  11 ms
>
```

By entering a **whois** command on a Unix device, the output shows that the route is as follows:

1. **192/8**: The local network of the AnywhereUSB Plus device.
2. **192.168.8.1**: The local network gateway to the Internet.
3. **96/8**: Charter Communications, the network provider.
4. **216/8**: Google Inc.

Stop the traceroute process

To stop the traceroute process, enter **Ctrl-C**.

File system

This chapter contains the following topics:

The AnywhereUSB Plus local file system

The AnywhereUSB Plus local file system has approximately 150 MB of space available for storing files, such as Python programs, alternative configuration files and firmware versions, and release files, such as cellular module images. The writable directories within the filesystem are:

- /tmp
- /opt
- /etc/config

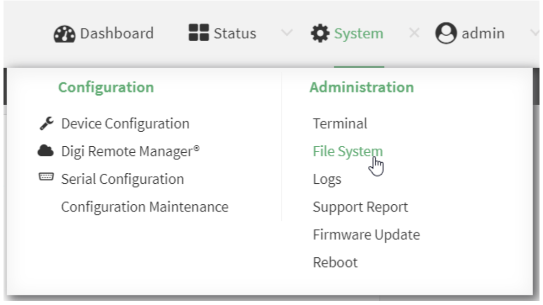
Files stored in the /tmp directory do not persist across reboots. Therefore, /tmp is a good location to upload temporary files, such as files used for firmware updates. Files stored in /opt and /etc/config do persist across reboots, but are deleted if a factory reset of the system is performed. See Erase device configuration and reset to factory defaults for more information.

Display directory contents

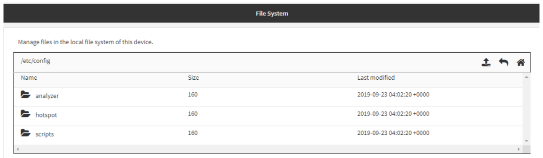
To display directory contents by using the WebUI or the Admin CLI:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **System**. Under **Administration**, click **File System**.



The **File System** page appears.



File System

Manage files in the local file system of this device.

/etc/config

Name	Size	Last modified
analyzer	160	2019-09-23 04:02:20 +0000
hotspot	160	2019-09-23 04:02:20 +0000
scripts	160	2019-09-23 04:02:20 +0000

3. Highlight a directory and click ↪ to open the directory and view the files in the directory.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type **ls** ***/path/dir_name***. For example, to display the contents of the **/etc/config** directory:

```
> ls /etc/config
-rw-r--r--    1 root     root           856 Nov 20 20:12 accns.json
drw-------    2 root     root           160 Sep 23 04:02 analyzer
drwxr-xr-x    3 root     root           224 Sep 23 04:02 cc_acl
-rw-r--r--    1 root     root            47 Sep 23 04:02 dhcp.leases
...
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Create a directory

Command line

This procedure is not available through the WebUI. To make a new directory, use the mkdir command, specifying the name of the directory.

For example:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type **mkdir */path/dir_name***. For example, to create a directory named **temp** in **/etc/config**:

```
> mkdir /etc/config/temp
>
```

3. Verify that the directory was created:

```
> ls /etc/config
...
-rw-r--r--    1 root     root          1436 Aug 12 21:36 ssl.crt
-rw-------    1 root     root          3895 Aug 12 21:36 ssl.pem
-rw-r--r--    1 root     root            10 Aug  5 06:41 start
drwxr-xr-x    2 root     root           160 Aug 25 17:49 temp
>
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Display file contents

This procedure is not available through the WebUI. To display the contents of a file by using the Admin CLI, , use the more command, specifying the name of the directory.

For example:

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the Admin CLI prompt, type **more *[path/filename*](*/path/filename*)**. For example, to view the contenct of the file **accns.json** in **/etc/config**:

```
> more /etc/config/accns.json
{
    "auth":
        "user": {
            "admin": {
                "password":
"$2a$05$W1sls1oxsadf/n4J0XT.Rgr6ewr1yerHtXQdbafsatGswKg0YUm"
            }
        }
    },
    "schema": {
        "version": "461"
    }
}
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Copy a file or directory

This procedure is not available through the WebUI. To copy a file or directory by using the Admin CLI, use the cp command, specifying the existing path and filename followed by the path and filename of the new file, or specifying the existing path and directory name followed by the path and directory name of the new directory.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the Admin CLI prompt, type **cp */path/filename*|*dir_name* */path*[*filename*]|*dir_name***. For example:

- To copy the file **/etc/config/accns.json** to a file named **backup_cfg.json** in a directory named **/etc/config/test**, enter the following:

  ```
  > cp /etc/config/accns.json /etc/config/test/backup_cfg.json
  >
  ```

- To copy a directory named **/etc/config/test** to **/opt**:

  ```
  > cp /etc/config/test/ /opt/
  >
  ```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Move or rename a file or directory

This procedure is not available through the WebUI. To move or rename a file or directory by using the Admin CLI, use the mv command.

Command line

To rename a file named **test.py** in **/etc/config/scripts** to **final.py**:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type:

   ```
   > mv /etc/config/scripts/test.py /etc/config/scripts/final.py
   >
   ```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

To move **test.py** from **/etc/config/scripts** to **/opt**:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type:

   ```
   > mv /etc/config/scripts/test.py /opt/
   >
   ```

3. Type **exit** to exit the Admin CLI.

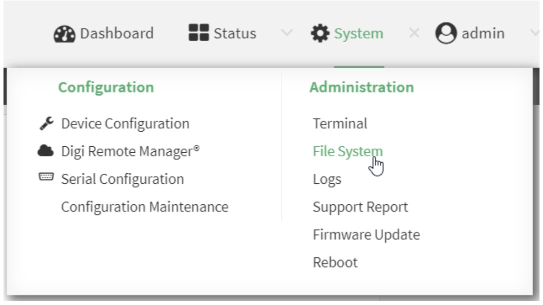
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Delete a file or directory

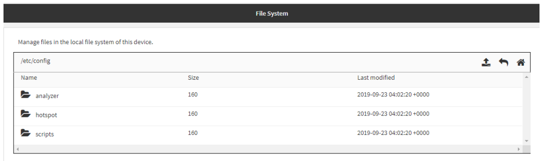
To delete a file or directory by using the WebUI or the Admin CLI:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **System**. Under **Administration**, click **File System**.



The **File System** page appears.



File System

Manage files in the local file system of this device.

/etc/config

Name	Size	Last modified
analyzer	160	2019-09-23 04:02:20 +0000
hotspot	160	2019-09-23 04:02:20 +0000
scripts	160	2019-09-23 04:02:20 +0000

3. Highlight the directory containing the file to be deleted and click ↪ to open the directory.
4. Highlight the file to be deleted and click 🗑.
5. Click **OK** to confirm.

Command line

To delete a file named **test.py** in **/etc/config/scripts**:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type:

```
> rm /etc/config/scripts/test.py
rm: remove '/etc/config/scripts/test.py'? yes
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

To delete a directory named **temp** from **/opt**:

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the Admin CLI prompt, type:

```
> rm /opt/temp/
rm: descend into directory '/opt/temp'? yes
rm: remove directory '/opt/temp'? yes
>
```

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

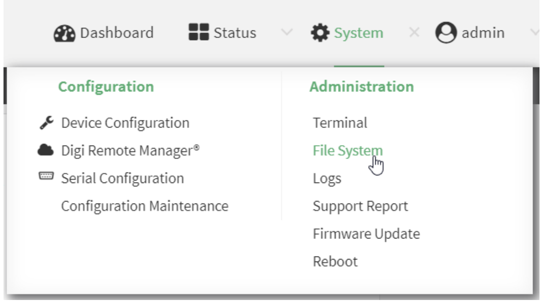
Upload and download files

You can download and upload files by using the WebUI or from the command line by using the scp Secure Copy command, or by using a utility such as SSH File Transfer Protocol (SFTP) or an SFTP application like FileZilla.

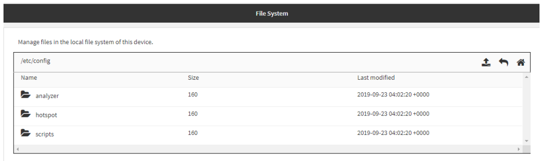
Upload and download files by using the WebUI

Upload files

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **System**. Under **Administration**, click **File System**.



   The **File System** page appears.



Manage files in the local file system of this device.

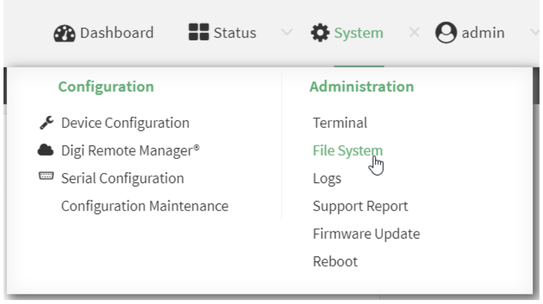
/etc/config

Name	Size	Last modified
analyzer	160	2019-09-23 04:02:20 +0000
hotspot	160	2019-09-23 04:02:20 +0000
scripts	160	2019-09-23 04:02:20 +0000

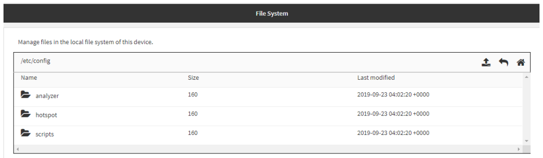
3. Highlight the directory to which the file will be uploaded and click ↩ to open the directory.
4. Click ⬆ (upload).
5. Browse to the location of the file on your local machine. Select the file and click **Open** to upload the file.

Download files

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **System**. Under **Administration**, click **File System**.



The **File System** page appears.



File System

Manage files in the local file system of this device.

/etc/config

Name	Size	Last modified
analyzer	160	2019-09-23 04:02:20 +0000
hotspot	160	2019-09-23 04:02:20 +0000
scripts	160	2019-09-23 04:02:20 +0000

3. Highlight the directory to which the file will be uploaded and click ➦ to open the directory.
4. Highlight the appropriate file and click ⤓ (download).

Upload and download files by using the Secure Copy command

Copy a file from a remote host to the AnywhereUSB Plus device

To copy a file from a remote host to the AnywhereUSB Plus device, use the scp command as follows:

```
> scp host hostname-or-ip user username remote remote-path local local-path to
local
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the path and filename of the file on the remote host that will be copied to the AnywhereUSB Plus device.
- *local-path* is the location on the AnywhereUSB Plus device where the copied file will be placed.

Transfer a file from the AnywhereUSB Plus device to a remote host

To copy a file from the AnywhereUSB Plus device to a remote host, use the scp command as follows:

```
> scp host hostname-or-ip user username remote remote-path local local-path to
remote
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the location on the remote host where the file will be copied.
- *local-path* is the path and filename on the AnywhereUSB Plus device.

To copy a support report from the AnywhereUSB Plus device to a remote host at the IP address of 192.168.4.1:

1. Use the **system support-report** command to generate the report:

```
> system support-report /var/log/
Saving support report to /var/log/support-report-0040D0133536-21-02-26-
8:04:23.bin
Support report saved.
>
```

2. Use the **scp** command to transfer the report to a remote host:

```
> scp host 192.168.4.1 user admin remote /home/admin/temp/ local
/var/log/support-report-00:40:D0:13:35:36-21-02-26-8:04:23.bin to remote
admin@192.168.4.1's password: adminpwd
support-report-0040D0133536-21-02-26-8:04:23.bin
>
```

Upload and download files using SFTP

Transfer a file from a remote host to the AnywhereUSB Plus device

This example uploads firmware from a remote host to the AnywhereUSB Plus device with an IP address of **192.168.2.1**, using the username **ahmed**:

```
$ sftp ahmed@192.168.2.1
Password:
Connected to 192.168.2.1
sftp> put AnywhereUSB Plus-21.2.39.67
Uploading AnywhereUSB Plus-21.2.39.67 to AnywhereUSB Plus-21.2.39.67
AnywhereUSB Plus-21.2.39.67
               100%  24M  830.4KB/s   00:00
sftp> exit
$
```

Transfer a file from the AnywhereUSB Plus device to a remote host

This example downloads a file named **test.py** from the AnywhereUSB device at the IP address of **192.168.2.1** with a username of **ahmed** to the local directory on the remote host:

```
$ sftp ahmed@192.168.2.1
Password:
Connected to 192.168.2.1
sftp> get test.py
Fetching test.py to test.py
test.py
  100%  254    0.3KB/s   00:00
```

```
sftp> exit
$
```

Routing

This chapter contains the following topics:

IP routing

The AnywhereUSB Plus device uses IP routes to decide where to send a packet it receives for a remote network. The process for deciding on a route to send the packet is as follows:

1. The device examines the destination IP address in the IP packet, and looks through the IP routing table to find a match for it.
2. If it finds a route for the destination, it forwards the IP packet to the configured IP gateway or interface.
3. If it cannot find a route for the destination, it uses a default route.
4. If there are two or more routes to a destination, the device uses the route with the longest mask.
5. If there are two or more routes to a destination with the same mask, the device uses the route with the lowest metric.

This section contains the following topics:

Configure a static route

A static route is a manually configured routing entry. Information about the route is manually entered rather than obtained from dynamic routing traffic.

Required configuration items

- The destination address or network.
- The interface to use to reach the destination.

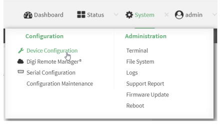
Additional configuration items

- A label used to identify this route.
- The IPv4 address of the gateway used to reach the destination.
- The metric for the route. When multiple routes are available to reach the same destination, the route with the lowest metric is used.
- The Maximum Transmission Units (MTU) of network packets using this route.

To configure a static route:

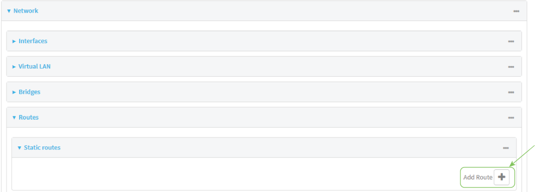
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

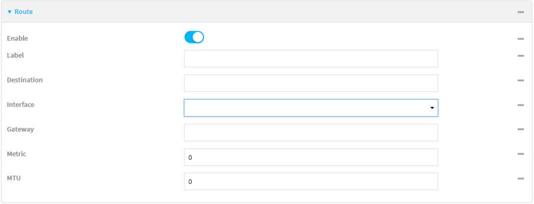


The **Configuration** window is displayed.

3. Click **Network** > **Routes** > **Static routes**.
4. Click the ✚ to add a new static route.



The new static route configuration page is displayed:

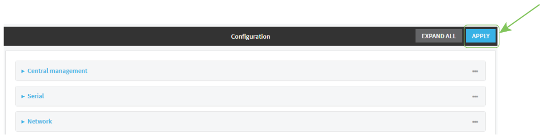


New static route configurations are enabled by default. To disable, click to toggle **Enable** to off.

5. (Optional) For **Label**, type a label that will be used to identify this route.

6. For **Destination**, type the IP address or network of the destination of this route.

   For example, to route traffic to the 192.168.47.0 network that uses a subnet mask of 255.255.255.0, type **192.168.47.0/24**. The **any** keyword can also be used to route packets to any destination with this static route.
7. For **Interface**, select the interface on the AnywhereUSB Plus device that will be used with this static route.
8. (Optional) For **Gateway**, type the IPv4 address of the gateway used to reach the destination. Set to blank if the destination can be accessed without a gateway.
9. (Optional) For **Metric**, type the metric for the route. When multiple routes are available to reach the same destination, the route with the lowest metric is used.
10. (Optional) For **MTU**, type the Maximum Transmission Units (MTU) of network packets using this route.
11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a new static route:

```
(config)> add network route static end
(config network route static 0)>
```

   New static route instances are enabled by default. To disable:

```
(config network route static 0)> enable false
(config network route static 0)>
```

4. (Optional) set a label that will be used to identify this route. For example:

```
(config network route static 0)> label "route to accounting network"
(config network route static 0)>
```

5. Set the IP address or network of the destination of this route. For example:

```
(config network route static 0)> destination ip_address[/netmask]
(config network route static 0)>
```

   For example, to route traffic to the 192.168.47.0 network that uses a subnet mask of 255.255.255.0:

```
(config network route static 0)> dst 192.168.47.0/24
(config network route static 0)>
```

The **any** keyword can also be used to route packets to any destination with this static route.

6. Set the interface on the AnywhereUSB Plus device that will be used with this static route:
    a. Use the **?** to determine available interfaces:

    ```
    (config network route static 0)> interface ?

    Interface: The network interface to use to reach the destination.
    Format:
      /network/interface/defaultip
      /network/interface/defaultlinklocal
      /network/interface/eth1
      /network/interface/eth2
      /network/interface/loopback
    Current value:

    (config network route static 0)> interface
    ```

    b. Set the interface. For example:

    ```
    (config network route static 0)> interface /network/interface/eth1
    (config network route static 0)>
    ```

7. (Optional) Set the IPv4 address of the gateway used to reach the destination. Set to blank if the destination can be accessed without a gateway.

    ```
    (config network route static 0)> gateway IPv4_address
    (config network route static 0)>
    ```

8. (Optional) Set the metric for the route. When multiple routes are available to reach the same destination, the route with the lowest metric is used.

    ```
    (config network route static 0)> metric value
    (config network route static 0)>
    ```

    where *value* is an interger between **0** and **65535**. The default is **0**.

9. (Optional) Set the Maximum Transmission Units (MTU) of network packets using this route:

    ```
    (config network route static 0)> mtu integer
    (config network route static 0)>
    ```

10. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

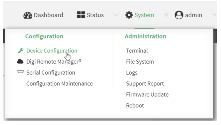
11. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

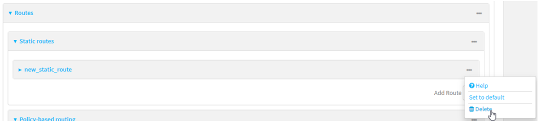
Delete a static route

WebUI

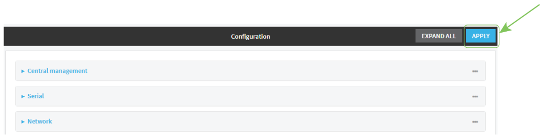
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Network** > **Routes** > **Static routes**.
4. Click the menu icon (**...**) for a static route and select **Delete**.

5. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Determine the index number of the static route to be deleted:

   ```
   (config)> show network route static
   0
       dst 10.0.0.1
       enable true
       no gateway
       interface /network/interface/lan1
       label new_static_route
       metric 0
       mtu 0
   1
   ```

```
    dst 192.168.5.1
    enable true
    gateway 192.168.5.1
    interface /network/interface/lan2
    label new_static_route_1
    metric 0
    mtu 0
(config)>
```

4. Use the index number to delete the static route:

```
(config)> del network route static 0
(config)>
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Policy-based routing

Normally, a routing device determines how to route a network packet based on its destination address. However, you can use policy-based routing to forward the packet based on other criteria, such as the source of the packet. For example, you can configure the AnywhereUSB Plus device so that high-priority traffic is routed through the cellular connection, while all other traffic is routed through an Ethernet (WAN) connection.

Policy-based routing for the AnywhereUSB Plus device uses the following criteria to determine how to route traffic:

- Firewall zone (for example, internal/outbound traffic, external/inbound traffic, or IPSec tunnel traffic).
- Network interface (for example, the cellular connection, the WAN, or the LAN).
- IPv4 address.
- IPv6 address.
- MAC address.
- Domain.
- Protocol type (TCP, UDP, ICMP, or all).

The order of the policies is important. Routing policies are processed sequentially; as a result, if a packet matches an earlier policy, it will be routed using that policy's rules. It will not be processed by any subsequent rules.

Configure a routing policy

Required configuration items

- The packet matching parameters. It can any combination of the following:
  - Source interface.
  - Source address. This can be a firewall zone, an interface, a single IPv4/IPv6 address or network, or a MAC address.
  - Destination address. This can be a firewall zone, an interface, a single IPv4/IPv6 address or network, or a domain.
  - Protocol. This can be **any**, **tcp**, **udp** or **icmp**.
  - Source port. This is only used if the protocol is set to **tcp** or **udp**.
  - Destination port. This is only used if protocol is set to **tcp** or **udp**.
- The network interface used to reach the destination.

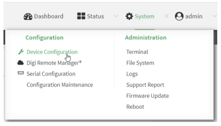
Additional configuration items

- A label for the routing policy.
- Whether packets that match this policy should be dropped when the gateway interface is disconnected, rather than forwarded through other interfaces.

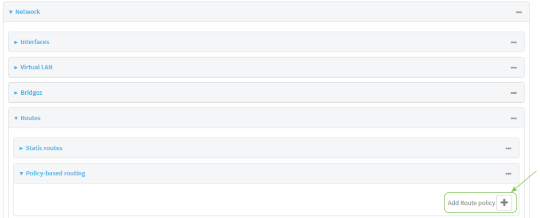
To configure a routing policy:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.
3. Click **Network** > **Routes** > **Policy-based routing**.
4. Click the ✚ to add a new route policy.



   The new route policy page is displayed:

   New route policies are enabled by default. To disable, click to toggle **Enable** to off.
5. (Optional) For **Label**, type a label that will be used to identify this route policy.

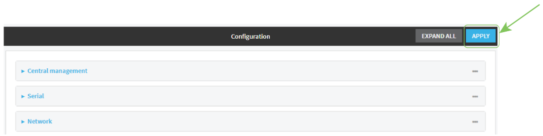
6. For **Interface**, select the interface on the AnywhereUSB Plus device that will be used with this route policy.
7. (Optional) Enable **Exclusive** to configure the policy to drop packets that match the policy when the gateway interface is disconnected, rather than forwarded through other interfaces.
8. For **IP version**, select **Any**, **IPv4**, or **IPv6**.
9. For **Protocol**, select **Any**, **TCP**, **UDP**, or **ICMP**.
   - If **TCP** or **UDP** is selected for **Protocol**, type the port numbers of the **Source port** and **Destination port**, or set to **any** to match for any port.
   - If **ICMP** is selected for **Protocol**, type the ICMP type and optional code, or set to **any** to match for any ICMP type.
10. For **DSCP**, type the 6-bit hexadecimal Differentiated Services Code Point (DSCP) field match criteria. This will match packets based on the DHCP field within the ToS field of the IP header.
11. Configure source address information:
    a. Click to expand **Source address**.
    b. For **Type**, select one of the following:
       - **Zone**: Matches the source IP address to the selected firewall zone. See Firewall configuration for more information about firewall zones.
       - **Interface**: Matches the source IP address to the selected interface's network address.
       - **IPv4 address**: Matches the source IP address to the specified IP address or network. Use the format ***IPv4_address***[/***netmask*]**, or use **any** to match any IPv4 address.
       - **IPv6 address**: Matches the source IP address to the specified IP address or network. Use the format ***IPv6_address*[/*prefix_length*]**, or use **any** to match any IPv6 address.
       - **MAC address**: Matches the source MAC address to the specified MAC address.
12. Configure the destination address information:
    a. Click to expand **Destination address**.
    b. For **Type**, select one of the following:
       - **Zone**: Matches the destination IP address to the selected firewall zone. See Firewall configuration for more information about firewall zones.
       - **Interface**: Matches the destination IP address to the selected interface's network address.
       - **IPv4 address**: Matches the destination IP address to the specified IP address or network. Use the format *IPv4_address*/[*netmask*], or use **any** to match any IPv4 address.
       - **IPv6 address**: Matches the destination IP address to the specified IP address or network. Use the format *IPv6_address*/[*prefix_length*], or use **any** to match any IPv6 address.
       - **Domain**: Matches the destination IP address to the specified domain names. To specify domains:
         i. Click to expand **Domains**.
         ii. Click the ✚ to add a domain.

iii. For **Domain**, type the domain name.

iv. Repeat to add additional domains.

- **Default route**: Matches packets destined for the default route, excluding routes for local networks.

13. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a new routing policy:

```
(config)> add network route policy end
(config network route policy 0)>
```

   New route policies are enabled by default. To disable:

```
(config network route policy 0)> enable false
(config network route policy 0)>
```

4. (Optional) Set the label that will be used to identify this route policy:

```
(config network route policy 0)> label "New route policy"
(config network route policy 0)>
```

5. Set the interface on the AnywhereUSB Plus device that will be used with this route policy:

   a. Use the **?** to determine available interfaces:

```
(config network route policy 0)> interface ?

Interface: The network interface used to reach the destination. Packets
that satisfy the matching criteria will be routed through this
interface. If the interface has a gateway then it will be used as the
next hop.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
```

```
Current value:

(config network route policy 0)> interface
```

b. Set the interface. For example:

```
(config network route policy 0)> interface /network/interface/eth1
(config network route policy 0)>
```

6. (Optional) Enable **exclusive** to configure the policy to drop packets that match the policy when the gateway interface is disconnected, rather than forwarded through other interfaces:

```
(config network route policy 0)> exclusive true
(config network route policy 0)>
```

7. Select the IP version:

```
(config network route policy 0)> ip_version value
(config network route policy 0)>
```

   where *value* is one of **any**, **ipv4**, or **ipv6**.

8. Set the protocol:

```
(config network route policy 0)> protocol value
(config network route policy 0)>
```

   where *value* is one of:

   - **any**: All protocols are matched.
   - **tcp**: Source and destination ports are matched:

     a. Set the source port:

```
(config network route policy 0)> src_port value
(config network route policy 0)>
```

     where *value* is the port number, or the keyword **any** to match any port as the source port.

     b. Set the destination port:

```
(config network route policy 0)> dst_port value
(config network route policy 0)>
```

     where *value* is the port number, or the keyword **any** to match any port as the destination port.

   - **upd**: Source and destination ports are matched:

     a. Set the source port:

```
(config network route policy 0)> src_port value
(config network route policy 0)>
```

     where *value* is the port number, or the keyword **any** to match any port as the source port.

b. Set the destination port:

```
(config network route policy 0)> dst_port value
(config network route policy 0)>
```

where *value* is the port number, or the keyword **any** to match any port as the destination port.

- **icmp**: The ICMP protocol is matched. Identify the ICMP type:

```
(config network route policy 0)> icmp_type value
(config network route policy 0)>
```

where *value* is the ICMP type and optional code, or set to **any** to match for any ICMP type.

9. Set the source address type:

```
(config network route policy 0)> src type value
(config network route policy 0)>
```

where *value* is one of:

- **zone**: Matches the source IP address to the selected firewall zone. Set the zone:

a. Use the **?** to determine available zones:

```
(config network route policy 0)> src zone ?

Zone: Match the IP address to the specified firewall zone.
Format:
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

Default value: any
Current value: any

(config network route policy 0)> src zone
```

b. Set the zone. For example:

```
(config network route policy 0)> src zone external
(config network route policy 0)>
```

See Firewall configuration for more information about firewall zones.

- **interface**: Matches the source IP address to the selected interface's network address. Set the interface:

a. Use the **?** to determine available interfaces:

```
(config network route policy 0)> src interface ?

Interface: The network interface.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config network route policy 0)> src interface
```

b. Set the interface. For example:

```
(config network route policy 0)> src interface
/network/interface/eth1
(config network route policy 0)>
```

- **address**: Matches the source IPv4 address to the specified IP address or network. Set the address that will be matched:

  ```
  (config network route policy 0)> src address value
  (config network route policy 0)>
  ```

  where value uses the format ***IPv4_address***[/***netmask***], or **any** to match any IPv4 address.
- **address6**: Matches the source IPv6 address to the specified IP address or network. Set the address that will be matched:

  ```
  (config network route policy 0)> src address6 value
  (config network route policy 0)>
  ```

  where value uses the format ***IPv6_address***[/***prefix_length***], or **any** to match any IPv6 address.
- **mac**: Matches the source MAC address to the specified MAC address. Set the MAC address to be matched:

  ```
  (config network route policy 0)> src mac MAC_address
  (config network route policy 0)>
  ```

10. Set the destination address type:

```
(config network route policy 0)> dst type value
(config network route policy 0)>
```

where *value* is one of:

- **zone**: Matches the destination IP address to the selected firewall zone. Set the zone:

  a. Use the **?** to determine available zones:

  ```
  (config network route policy 0)> dst zone ?

  Zone: Match the IP address to the specified firewall zone.
  Format:
    any
    dynamic_routes
    edge
    external
    internal
    ipsec
    loopback
    setup

  Default value: any
  Current value: any

  (config network route policy 0)> dst zone
  ```

  b. Set the zone. For example:

  ```
  (config network route policy 0)> dst zone external
  (config network route policy 0)>
  ```

  See Firewall configuration for more information about firewall zones.

- **interface**: Matches the destination IP address to the selected interface's network address. Set the interface:

  a. Use the **?** to determine available interfaces:

  ```
  (config network route policy 0)> dst interface ?

  Interface: The network interface.
  Format:
    /network/interface/defaultip
    /network/interface/defaultlinklocal
    /network/interface/eth1
    /network/interface/eth2
    /network/interface/loopback
  Current value:

  (config network route policy 0)> dst interface
  ```

  b. Set the interface. For example:

  ```
  (config network route policy 0)> dst interface
  /network/interface/eth1
  (config network route policy 0)>
  ```

- **address**: Matches the destination IPv4 address to the specified IP address or network. Set the address that will be matched:

  ```
  (config network route policy 0)> dst address value
  (config network route policy 0)>
  ```

  where value uses the format ***IPv4_address***[/***netmask***], or **any** to match any IPv4 address.
- **address6**: Matches the destination IPv6 address to the specified IP address or network. Set the address that will be matched:

  ```
  (config network route policy 0)> dst address6 value
  (config network route policy 0)>
  ```

  where value uses the format ***IPv6_address***[/***prefix_length***], or **any** to match any IPv6 address.
- **mac**: Matches the destination MAC address to the specified MAC address. Set the MAC address to be matched:

  ```
  (config network route policy 0)> dst mac MAC_address
  (config network route policy 0)>
  ```

11. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

12. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Routing services

Your AnywhereUSB Plus includes support for dynamic routing services and protocols. The following routing services are supported:

Service or protocol	Information
RIP	The IPv4 Routing Information Protocol (RIP) service supports RIPv2 (RFC2453) and RIPv1 (RFC1058).
RIPng	The IPv6 Routing Information Protocol (RIP) service supports RIPng (RFC2080).
OSPFv2	The IPv4 Open Shortest Path First (OSPF) service supports OSPFv2 (RFC2328).
OSPFv3	The IPv6 Open Shortest Path First (OSPF) service supports OSPFv3 (RFC2740).
BGP	The Border Gateway Protocol (BGP) service supports BGP-4 (RFC1771).
Babel	The IPv4 and IPv6 Babel service.
IS-IS	The IPv4 and IPv6 Intermediate System to Intermediate System (IS-IS) service.

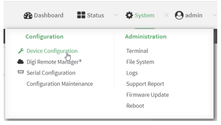
Configure routing services

Required configuration items

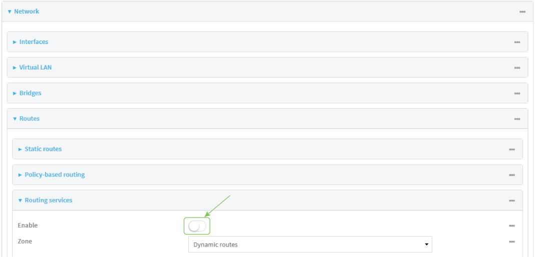
- Enable routing services.
- Enable and configure the types of routing services that will be used.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

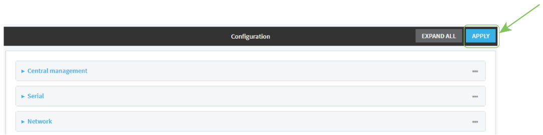
   The **Configuration** window is displayed.
3. Click **Network** > **Routes** > **Routing services**.
4. Click **Enable**.

   Enable

   Zone: Dynamic routes

   The default firewall zone setting, **Dynamic routes**, is specifically designed to work with routing services and should be left as the default.
5. Configure the routing services that will be used:
   a. Click to expand a routing service.
   b. **Enable** the routing service.
   c. Complete the configuration of the routing service.
6. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Enable routing services:

```
(config)> network route service enable true
(config)>
```

4. Configure routing services that will be used:
   a. Use the **?** to display available routing services:

```
(config)> network route service ?

Routing services: Settings for dynamic routing services and protocols.

 Parameters                Current Value
 -----------------------------------------------------------------------
--------
 enable                    true            Enable
 zone                      dynamic_routes  Zone

 Additional Configuration
 -----------------------------------------------------------------------
--------
 babel                     Babel
 bgp                       BGP
 isis                      IS-IS
 ospfv2                    OSPFv2
 ospfv3                    OSPFv3
 rip                       RIP
 ripng                     RIPng

(config)>
```

   b. Enable a routing service that will be used. For example, to enable the RIP service:

```
(config)> network route service rip enable true
(config)>
```

   c. Complete the configuration of the routing service. For example, use the **?** to view the available parameters for the RIP service:

```
(config)> network route service rip ?

 Parameters                Current Value
 -----------------------------------------------------------------------
--------
 ecmp                      false          Allow ECMP
 enable                    true           Enable

 Additional Configuration
 -----------------------------------------------------------------------
--------
 interface                 Interfaces
 neighbour                  Neighbours
 redis                      Route redistribution
 timer                      Timers

(config)>
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

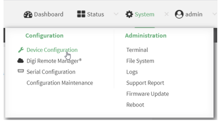
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show the routing table

To display the routing table:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Status** > **Routes**.

   The **Network Routing** window is displayed.
4. Click **IPv4 Load Balance** to view IPv4 load balancing.
5. Click **IPv6 Load Balance** to view IPv6 load balancing.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type show route:

   You can limit the display to only IPv4 entries by using **show route ipv4**, or to IPv6 entries by using **show route ipv6**. You can also display more information by adding the **verbose** option to the **show route** and **show route *ip_type*** commands.
3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Dynamic DNS

The Domain Name System (DNS) uses name servers to provide a mapping between computer-readable IP addresses and human-readable hostnames. This allows users to access websites and personal networks with easy-to-remember URLs. Unfortunately, IP addresses change frequently, invalidating these mappings when they do. Dynamic DNS has become the standard method of addressing this problem, allowing devices to update name servers with their new IP addresses.

By providing the AnywhereUSB Plus device with the domain name and credentials obtained from a dynamic DNS provider, the router can automatically update the remote nameserver whenever your WAN or public IP address changes.

Your AnywhereUSB Plus device supports a number of Dynamic DNS providers as well as the ability to provide a custom provider that is not included on the list of providers.

Configure dynamic DNS

This section describes how to cofigure dynamic DNS on a AnywhereUSB Plus device.

Required configuration items

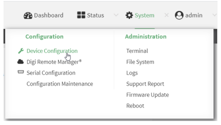
- Add a new Dynamic DNS service.
- The interface that has its IP address registered with the Dynamic DNS provider.
- The name of a Dynamic DNS provider.
- The domain name that is linked to the interface's IP address.
- The username and password to authenticate with the Dynamic DNS provider.

Additional configuration items

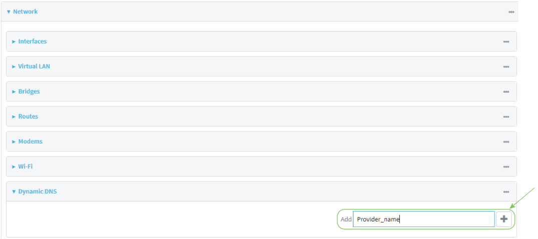
- If the Dynamic DNS service provider is set to **custom**, identify the URL that should be used to update the IP address with the Dynamic DNS provider.
- The amount of time to wait to check if the interface's IP address needs to be updated.
- The amount of time to wait to force an update of the interface's IP address.
- The amount of time to wait for an IP address update to succeed before retrying the update.
- The number of times to retry a failed IP address update.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

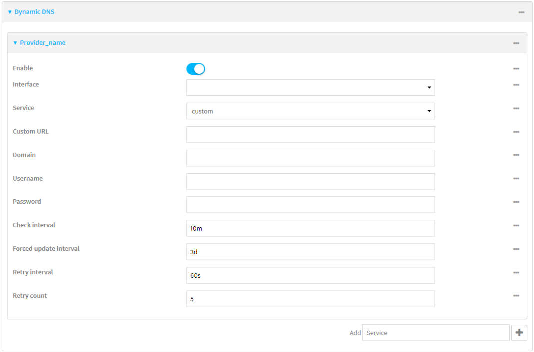
   

   The **Configuration** window is displayed.
3. Click **Network** > **Dynamic DNS**.
4. Type a name for this Dynamic DNS instance in **Add Service** and click ✚.

   Add Provider_name

   The Dynamic DNS configuration page displays.

   Enable ●
   Interface
   Service custom
   Custom URL
   Domain
   Username
   Password
   Check interval 10m
   Forced update interval 3d
   Retry interval 60s
   Retry count 5
   Add Service ✚

   New Dynamic DNS configurations are enabled by default. To disable, click to toggle **Enable** to off.
5. For **Interface**, select the interface that has its IP address registered with the Dynamic DNS provider.
6. For **Service**, select the Dynamic DNS provider, or select **custom** to enter a custom URL for the Dynamic DNS provider.
7. If **custom** is selected for **Service**, type the **Custom URL** that should be used to update the IP address with the Dynamic DNS provider.
8. Type the **Domain** name that is linked to the interface's IP address.
9. Type the **Username** and **Password** used to authenticate with the Dynamic DNS provider.
10. (Optional) For **Check Interval**, type the amount of time to wait to check if the interface's IP address needs to be updated.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number*****{w|d|h|m|s}**.

For example, to set **Check interval** to ten minutes, enter **10m** or **600s**.

11. (Optional) For **Forced update interval**, type the amount of time to wait to force an update of the interface's IP address.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **Forced update interval** to ten minutes, enter **10m** or **600s**.

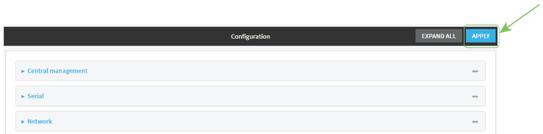
    The setting for **Forced update interval** must be larger than the setting for **Check Interval**.

12. (Optional) For **Retry interval**, type the amount of time to wait for an IP address update to succeed before retrying the update.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **Retry interval** to ten minutes, enter **10m** or **600s**.

13. (Optional) For **Retry count**, type the number of times to retry a failed IP address update.
14. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a new Dynamic DNS instance. For example, to add an instance named **new_ddns_instance**:

```
(config)> add network ddns new_ddns_instance
(config network ddns new_ddns_instance)>
```

   New Dynamic DNS instances are enabled by default. To disable:

```
(config network ddns new_ddns_instance)> enable false
(config network ddns new_ddns_instance)>
```

4. Set the interface for the Dynamic DNS instance:
   a. Use the **?** to determine available interfaces:

```
(config network ddns new_ddns_instance)> interface ?

Interface: The network interface from which to obtain the IP address to
register with the dynamic DNS service.
```

```
Format:
  defaultip
  defaultlinklocal
  eth1
  eth2
  loopback
Current value:

(config network ddns new_ddns_instance)> interface
```

b. Set the interface. For example:

```
(config network ddns new_ddns_instance)> interface eth1
(config network ddns new_ddns_instance)>
```

5. Set the Dynamic DNS provider service:

   a. Use the **?** to determine available services:

```
(config network ddns new_ddns_instance)> service ?

Service: The provider of the dynamic DNS service.
Format:
  custom
  3322.org
  changeip.com
  ddns.com.br
  dnsdynamic.org
  ...

Default value: custom
Current value: custom

(config network ddns new_ddns_instance)> service
```

   b. Set the service:

```
(config network ddns new_ddns_instance)> service service_name
(config network ddns new_ddns_instance)>
```

6. If **custom** is configured for **service**, set the custom URL that should be used to update the IP address with the Dynamic DNS provider:

```
(config network ddns new_ddns_instance)> custom url
(config network ddns new_ddns_instance)>
```

7. Set the domain name that is linked to the interface's IP address:

```
(config network ddns new_ddns_instance)> domain domain_name
(config network ddns new_ddns_instance)>
```

8. Set the username to authenticate with the Dynamic DNS provider:

   ```
   (config network ddns new_ddns_instance)> username name
   (config network ddns new_ddns_instance)>
   ```

9. Set the password to authenticate with the Dynamic DNS provider:

   ```
   (config network ddns new_ddns_instance)> password pwd
   (config network ddns new_ddns_instance)>
   ```

10. (Optional) Set the amount of time to wait to check if the interface's IP address needs to be updated:

    ```
    (config network ddns new_ddns_instance)> check_interval value
    (config network ddns new_ddns_instance)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

    For example, to set **check_interval** to ten minutes, enter either **10m** or **600s**:

    ```
    (config network ddns new_ddns_instance)> check_interval 600s
    (config network ddns new_ddns_instance)>
    ```

    The default is **10m**.

11. (Optional) Set the amount of time to wait to force an update of the interface's IP address:

    ```
    (config network ddns new_ddns_instance)> force_interval value
    (config network ddns new_ddns_instance)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

    For example, to set **force_interval** to ten minutes, enter either **10m** or **600s**:

    ```
    (config network ddns new_ddns_instance)> force_interval 600s
    (config network ddns new_ddns_instance)>
    ```

    The default is **3d**.

12. (Optional) Set the amount of time to wait for an IP address update to succeed before retrying the update:

    ```
    (config network ddns new_ddns_instance)> retry_interval value
    (config network ddns new_ddns_instance)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

    For example, to set **retry_interval** to ten minutes, enter either **10m** or **600s**:

    ```
    (config network ddns new_ddns_instance)> retry_interval 600s
    (config network ddns new_ddns_instance)>
    ```

    The default is **60s**.

13. (Optional) Set the number of times to retry a failed IP address update:

```
(config network ddns new_ddns_instance)> retry_count value
(config network ddns new_ddns_instance)>
```

where *value* is any interger. The default is **5**.

14. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

15. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Virtual Router Redundancy Protocol (VRRP)

Virtual Router Redundancy Protocol (VRRP) is a standard for gateway device redundancy and failover that creates a "virtual router" with a floating IP address. Devices connected to the LAN then use this virtual router as their default gateway. Responsibility for the virtual router is assigned to one of the VRRP-enabled devices on a LAN (the "master router"), and this responsibility transparently fails over to backup VRRP devices if the master router fails. This prevents the default gateway from being a single point of failure, without requiring configuration of dynamic routing or router discovery protocols on every host.

Multiple AnywhereUSB Plus devices can be configured as VRRP devices and assigned a priority. The router with the highest priority will be used as the master router. If the master router fails, then the IP address of the virtual router is mapped to the backup device with the next highest priority. Each VRRP router is configured with a unique LAN IP address, and the same shared VRRP address.

VRRP+

VRRP+ is an extension to the VRRP standard that uses network probing to monitor connections through VRRP-enabled devices and can dynamically change the priority of the devices, including changing devices from master to backup, and from backup to master, even if the device has not failed. For example, if a host becomes unreachable on the far end of a network link, then the physical default gateway can be changed by adjusting the VRRP priority of the AnywhereUSB device connected to the failing link. This provides failover capabilities based on the status of connections behind the router, in addition to the basic VRRP device failover. For AnywhereUSB Plus devices, Surelink is used to probe network connections.

VRRP+ can be configured to probe a specified IP address by either sending an ICMP echo request (ping) or attempting to open a TCP socket to the IP address.

Configure VRRP

This section describes how to configure VRRP on a AnywhereUSB Plus device.

Required configuration items

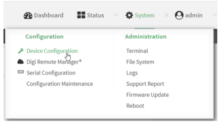
- Enable VRRP.
- The interface used by VRRP.

- The Router ID that identifies the virtual router instance. The Router ID must be the same on all VRRP devices that participate in the same VRRP device pool.
- The VRRP priority of this device.
- The shared virtual IP address for the VRRP virtual router. Devices connected to the LAN will use this virtual IP address as their default gateway.

See Configure VRRP+ for information about configuring VRRP+, an extension to VRRP that uses network probing to monitor connections through VRRP-enabled devices and dynamically change the VRRP priorty of devices based on the status of their network connectivity.

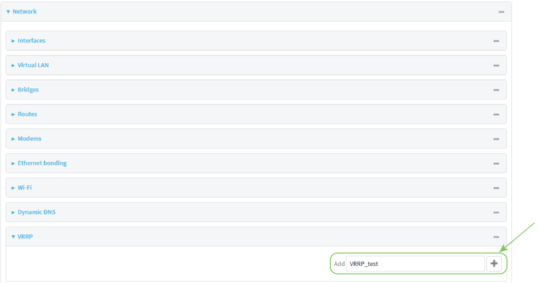
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



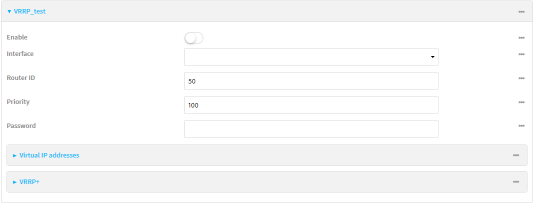
The **Configuration** window is displayed.

3. Click **Network** > **VRRP**.
4. For **Add VRRP instance**, type a name for the VRRP instance and click ✚.



Add VRRP_test

The new VRRP instance configuration is displayed.



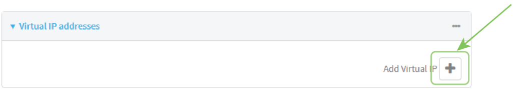
Router ID 50

Priority 100

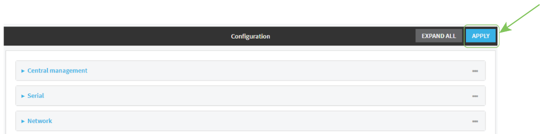
5. Click **Enable**.
6. For **Interface**, select the interface on which this VRRP instance should run.
7. For **Router ID** field, type the ID of the virtual router instance. The Router ID must be the same on all VRRP devices that participate in the same VRRP device pool. Allowed values are from **1** and **255**, and it is configured to **50** by default.
8. For **Priority**, type the priority for this router in the group. The router with the highest priority will be used as the master router. If the master router fails, then the IP address of the virtual

router is mapped to the backup device with the next highest priority. If this device's actual IP address is being used as the virtual IP address of the VRRP pool, then the priority of this device should be set to **255** . Allowed values are from **1** and **255**, and it is configured to **100** by default.

9. (Optional) For **Password**, type a password that will be used to authenticate this VRRP router with VRRP peers. If the password length exceeds 8 characters, it will be truncated to 8 characters.
10. Configure the virtual IP addresses associated with this VRRP instance:
    a. Click to expand **Virtual IP addresses**.
    b. Click ✚ to add a virtual IP address.



    c. For **Virtual IP**, type the IPv4 or IPv6 address for a virtual IP of this VRRP instance.
    d. (Optional) Repeat to add additional virtual IPs.
11. See Configure VRRP+ for information about configuring VRRP+.
12. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a VRRP instance. For example:

```
(config)> add network vrrp VRRP_test
(config network vrrp VRRP_test)>
```

4. Enable the VRRP instance:

```
(config network vrrp VRRP_test)> enable true
(config network vrrp VRRP_test)>
```

5. Set the interface on which this VRRP instance should run:
    a. Use the **?** to determine available interfaces:

```
(config network vrrp VRRP_test)> interface ?

Interface: The network interface to communicate with VRRP peers on and
```

```
listen for traffic to virtual IP addresses.
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config network vrrp VRRP_test)> interface
```

   b. Set the interface, for example:

```
(config network vrrp VRRP_test)> interface /network/interface/eth2
(config network vrrp VRRP_test)>
```

   c. Repeat for additional interfaces.

6. Set the router ID. The Router ID must be the same on all VRRP devices that participate in the same VRRP device pool. Allowed values are from **1** and **255**, and it is configured to **50** by default.

```
(config network vrrp VRRP_test)> router_id int
(config network vrrp VRRP_test)>
```

7. Set the priority for this router in the group. The router with the highest priority will be used as the master router. If the master router fails, then the IP address of the virtual router is mapped to the backup device with the next highest priority. If this device's actual IP address is being used as the virtual IP address of the VRRP pool, then the priority of this device should be set to **255** . Allowed values are from **1** and **255**, and it is configured to **100** by default.

```
(config network vrrp VRRP_test)> priority int
(config network vrrp VRRP_test)>
```

8. (Optional) Set a password that will be used to authenticate this VRRP router with VRRP peers. If the password length exceeds 8 characters, it will be truncated to 8 characters.

```
(config network vrrp VRRP_test)> password pwd
(config network vrrp VRRP_test)>
```

9. Add a virtual IP address associated with this VRRP instance. This can be an IPv4 or IPv6 address.

```
(config network vrrp VRRP_test)> add virtual_address end ip_address
(config network vrrp VRRP_test)>
```

   Additional virtual IP addresses can be added by repeating this step with different values for *ip_address*.

10. Save the configuration and apply the change:

```
(config network vrrp new_vrrp_instance)> save
Configuration saved.
>
```

11. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure VRRP+

VRRP+ is an extension to the VRRP standard that uses SureLink network probing to monitor connections through VRRP-enabled devices and adjust devices' VRRP priority based on the status of the SureLink tests.

This section describes how to configure VRRP+ on a AnywhereUSB Plus device.

Required configuration items

- Both master and backup devices:
    - A configured and enabled instance of VRRP. See Configure VRRP for information.
    - Enable VRRP+.
    - WAN interfaces to be monitored by using VRRP+.

      Note SureLink is enabled by default on all WAN interfaces, and should not be disabled on the WAN interfaces that are being monitored by VRRP+.

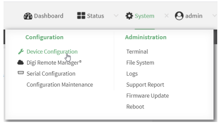
      If multiple WAN interfaces are being monitored on the same device, the VRRP priority will be adjusted only if all WAN interfaces fail SureLink tests.
    - The amount that the VRRP priority will be modified when SureLink determines that the VRRP interface is not functioning correctly.
    - Configure the VRRP interface's DHCP server to use a custom gateway that corresponds to one of the VRRP virtual IP addresses.
- Backup devices only:
    - Enable and configure SureLink on the VRRP interface.
    - Set the IP gateway to the IP address of the VRRP interface on the master device.

Additional configuration items

- For backup VRRP devices, enable the ability to monitor the VRRP master, so that a backup device can increase its priority when the master device fails SureLink tests.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

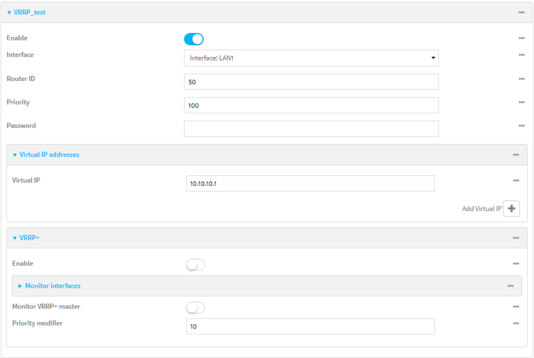


   The **Configuration** window is displayed.
3. Click **Network** > **VRRP**.

4. Create a new VRRP instance, or click to expand an existing VRRP instance.

   See Configure VRRP for information about creating a new VRRP instance.

5. Click to expand **VRRP+**.



▾ VRRP_test

Enable: ●

Interface: Interface: LAN1

Router ID: 50

Priority: 100

Password:

▾ Virtual IP addresses

Virtual IP: 10.10.10.1

Add Virtual IP ✚

▾ VRRP+

Enable: ○

▸ Monitor interfaces

Monitor VRRP+ master: ○

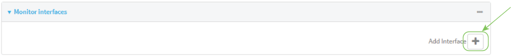
Priority modifier: 10

6. Click **Enable**.

7. Add interfaces to monitor:

   a. Click to expand **Monitor interfaces**.

   b. Click ✚ to add an interface for monitoring.



   c. For **Interface**, select the local interface to monitor. Generally, this will be a cellular or WAN interface.

   d. (Optional) Click ✚ again to add additional interfaces.

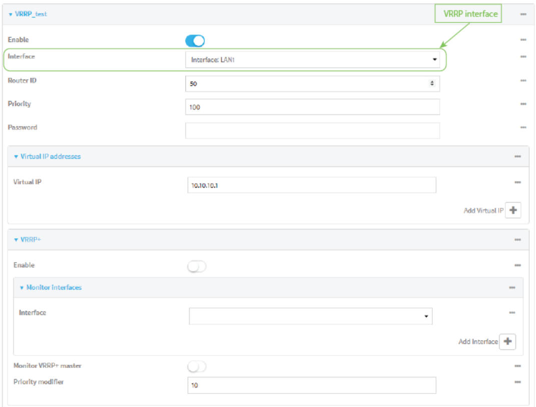
8. (Optional) For backup devices, click to enable **Monitor VRRP+ master**.

   This parameter allows a backup VRRP device to monitor the master device, and increase its priority when the master device is failing SureLink tests. This can allow a device functioning as a backup device to promote itself to master.

9. For **Priority modifier**, type or select the amount that the device's priority should be decreased due to SureLink connectivity failure, and increased when SureLink succeeds again.

   Along with the priority settings for devices in this VRRP pool, the amount entered here should be large enough to automatically demote a master device when SureLink connectivity fails. For example, if the VRRP master device has a priority of **100** and the backup device has a priority of **80**, then the **Priority modifier** should be set to an amount greater than **20** so that if SureLink fails on the master, it will lower its priority to below **80**, and the backup device will assume the master role.

10. Configure the VRRP interface. The VRRP interface is defined in the **Interface** parameter of the VRRP configuration, and generally should be a LAN interface:


VRRP interface

▾ VRRP_test

Enable: ●

Interface: Interface: LAN1

Router ID: 50

Priority: 100

Password:

▾ Virtual IP addresses

Virtual IP: 10.10.10.1

▾ VRRP+

Enable: ○

▾ Monitor Interfaces

Interface:

Monitor VRRP+ master: ○

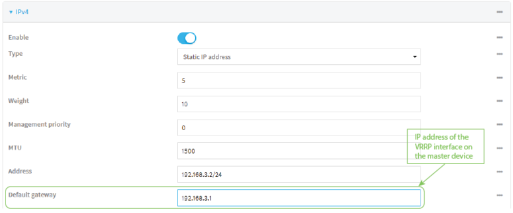
Priority modifier: 10

To configure the VRRP interface:

a. Click to expand **Network** > **Interfaces**.

b. Click to expand the appropriate VRRP interface (for example, **LAN1**).

c. For backup devices, for **Default Gateway**, type the IP address of the VRRP interface on the master device.


▾ IPv4

Enable: ●

Type: Static IP address

Metric: 5

Weight: 10

Management priority: 0

MTU: 1500

Address: 192.168.3.2/24

Default gateway: 192.168.3.1

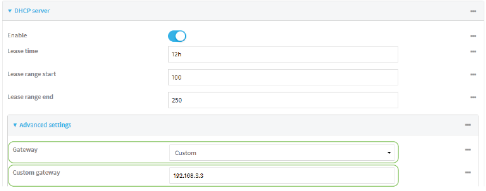
IP address of the VRRP interface on the master device

d. Configure the VRRP interface's DHCP server to use a custom gateway that corresponds to one of the VRRP virtual IP addresses:

   i. Click to expand **DHCP Server** > **Advanced settings**.

   ii. For **Gateway**, select **Custom**.

   iii. For **Custom gateway**, enter the IP address of one of the virtual IPs used by this VRRP instance.


▾ DHCP server

Enable: ●

Lease time: 12h

Lease range start: 100

Lease range end: 250

▾ Advanced settings

Gateway: Custom

Custom gateway: 192.168.3.3

e. For backup devices, enable and configure SureLink on the VRRP interface. Generally, this should be a LAN interface; VRRP+ will then monitor the LAN using SureLink to determine if the interface has network connectivity and promote a backup to master if SureLink fails.

   i. Click to expand **IPv4** > **SureLink**.

   ii. Click **Enable**.

   iii. For **Interval**, type a the amount of time to wait between connectivity tests. To guarantee seamless internet access for VRRP+ purposes, SureLink tests should occur more often than the default of 15 minutes.

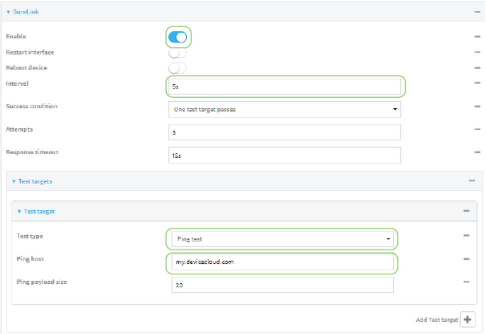
Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number*{w|d|h|m|s}**. For example, to set **Interval** to five seconds, enter **5s**.

iv. Click to expand **Test targets** > **Test target**.

v. Configure the test target. For example, to configure SureLink to verify internet connectivity on the LAN by pinging my.devicecloud.com:

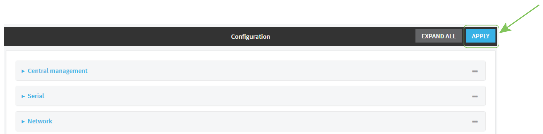
   i. For **Test Type**, select **Ping test**.

   ii. For **Ping host**, type **my.devicecloud.com**.



Enable: on
Restart interface: off
Reboot device: off
Interval: 5s
Success condition: One test target passes
Attempts: 3
Response timeout: 15s
Test type: Ping test
Ping host: my.devicecloud.com
Ping payload size: 20

11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Create a new VRRP instance, or edit an existing one. See Configure VRRP for information about creating a new VRRP instance.

4. Enable VRRP+:

```
(config)> network vrrp VRRP_test vrrp_plus enable true
(config)>
```

5. Add interfaces to monitor. Generally, this will be a cellular or WAN interface.

   a. Use the **?** to determine available interfaces:

```
(config)> network vrrp test interface ?

Interface: The network interface.
```

```
Format:
  /network/interface/defaultip
  /network/interface/defaultlinklocal
  /network/interface/eth1
  /network/interface/eth2
  /network/interface/loopback
Current value:

(config)> network vrrp test interface
```

b. Set the interface, for example:

```
(config)> add network vrrp VRRP_test vrrp_plus monitor_interface end
/network/interface/modem
(config)>
```

c. (Optional) Repeat for additional interfaces.

6. Set the amount that the device's priority should be decreased or increased due to SureLink connectivity failure or success:

   ```
   (config)> network vrrp VRRP_test vrrp_plus weight value
   (config)>
   ```

   where *value* is an integer between **1** and **254**. The default is **10**.

   Along with the priority settings for devices in this VRRP pool, the amount entered here should be large enough to automatically demote a master device when SureLink connectivity fails. For example, if the VRRP master device has a priority of **100** and the backup device has a priority of **80**, then **weight** should be set to an amount greater than **20** so that if SureLink fails on the master, it will lower its priority to below **80**, and the backup device will assume the master role.

7. (Optional) For backup devices, enable the ability for the device to monitor the master device. This allows a backup VRRP device to monitor the master device, and increase its priority when the master device is failing SureLink tests. This can allow a device functioning as a backup device to promote itself to master.

   ```
   (config)> network vrrp VRRP_test vrrp_plus monitor_master true
   (config)>
   ```

8. Configure the VRRP interface:

   a. Configure the VRRP interface's DHCP server to use a custom gateway that corresponds to one of the VRRP virtual IP addresses:

      i. Set the DHCP server gateway type to custom:

      ```
      (config)> network interface eth2 ipv4 dhcp_server advanced gateway
      custom
      (config)>
      ```

ii. Determine the VRRP virtual IP addresses:

```
(config)> show network vrrp VRRP_test virtual_address
0 192.168.3.3
1 10.10.10.1

(config)>
```

iii. Set the custom gateway to one of the VRRP virtual IP addresses. For example:

```
(config)> network interface eth2 ipv4 dhcp_server advanced gateway_
custom 192.168.3.3
(config)>
```

b. For backup devices, set the default gateway to the IP address of the VRRP interface on the master device. For example:

```
(config)> network interface eth2 ipv4 gateway 192.168.3.1
(config)>
```

c. For backup devices, enable and configure SureLink on the VRRP interface.

i. Determine the VRRP interface. Generally, this should be a LAN interface; VRRP+ will then monitor the LAN using SureLink to determine if the interface has network connectivity and promote a backup to master if SureLink fails.

```
(config)> show network vrrp VRRP_test interface
/network/interface/eth2
(config)>
```

ii. Enable SureLink on the interface:

```
(config)> network interface eth2 ipv4 surelink enable true
(config)>
```

iii. Set the amount of time to wait between connectivity tests:

```
(config)> network interface eth2 ipv4 surelink interval value
(config)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **interval** to ten minutes, enter **5s**:

```
(config)> network interface eth2 ipv4 surelink interval 5s
(config)>
```

iv. Create a SureLink test target:

```
(config)> add network interface eth2 ipv4 surelink target end
(config network interface eth2 ipv4 surelink target 0)>
```

v. Configure the type of test for the test target:

```
(config network interface eth2 ipv4 surelink target 0)> test value
(config network interface eth2 ipv4 surelink target 0)>
```

where *value* is one of:

- **ping**: Tests connectivity by sending an ICMP echo request to a specified hostname or IP address.
  - Specify the hostname or IP address:

    ```
    (config network interface eth2 ipv4 surelink target 0)>
    ping_host host
    (config network interface eth2 ipv4 surelink target 0)>
    ```

  - (Optional) Set the size, in bytes, of the ping packet:

    ```
    (config network interface eth2 ipv4 surelink target 0)>
    ping_size [num]
    (config network interface eth2 ipv4 surelink target 0)>
    ```

- **dns**: Tests connectivity by sending a DNS query to the specified DNS server.
  - Specify the DNS server. Allowed value is the IP address of the DNS server.

    ```
    (config network interface eth2 ipv4 surelinktarget 0)> dns_
    server ip_address
    (config network interface eth2 ipv4 surelinktarget 0)>
    ```

- **dns_configured**: Tests connectivity by sending a DNS query to the DNS servers configured for this interface.
- **http**: Tests connectivity by sending an HTTP or HTTPS GET request to the specified URL.
  - Specify the url:

    ```
    (config network interface eth2 ipv4 surelink target 0)>
    http_url value
    (config network interface eth2 ipv4 surelink target 0)>
    ```

    where *value* uses the format **http[s]://*hostname*/[*path*]**
- **interface_up**: The interface is considered to be down based on the interfaces down time, and the amount of time an initial connection to the interface takes before this test is considered to have failed.
  - (Optional) Set the amount of time that the interface can be down before this test is considered to have failed:

    ```
    (config network interface eth2 ipv4 surelink target 0)>
    interface_down_time value
    (config network interface eth2 ipv4 surelink target 0)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

For example, to set **interface_down_time** to ten minutes, enter either **10m** or **600s**:

```
(config network interface eth2 ipv4 surelink target 0)>
interface_down_time 600s
(config network interface eth2 ipv4 surelink target 0)>
```

The default is 60 seconds.

- (Optional) Set the amount of time to wait for an initial connection to the interface before this test is considered to have failed:

```
(config network interface eth2 ipv4 surelink target 0)>
interface_timeout value
(config network interface eth2 ipv4 surelink target 0)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number*****{w|d|h|m|s}**.

For example, to set **interface_timeout** to ten minutes, enter either **10m** or **600s**:

```
(config network interface eth2 ipv4 surelink target 0)>
interface_timeout 600s
(config network interface eth2 ipv4 surelink target 0)>
```

The default is 60 seconds.

9. Save the configuration and apply the change:

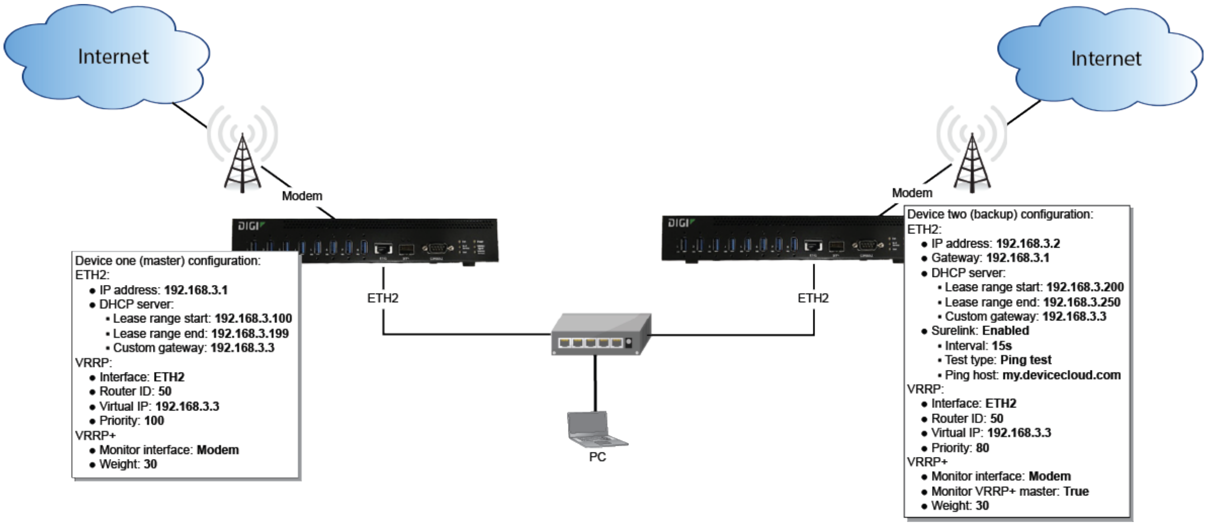
```
(config)> save
Configuration saved.
>
```

10. Type **exit** to exit the Admin CLI.

Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Example: VRRP/VRRP+ configuration

This example configuration creates a VRRP pool containing two AnywhereUSB Plus devices:



Device one (master) configuration:
ETH2:
- IP address: **192.168.3.1**
- DHCP server:
  - Lease range start: **192.168.3.100**
  - Lease range end: **192.168.3.199**
  - Custom gateway: **192.168.3.3**

VRRP:
- Interface: **ETH2**
- Router ID: **50**
- Virtual IP: **192.168.3.3**
- Priority: **100**

VRRP+
- Monitor interface: **Modem**
- Weight: **30**

Device two (backup) configuration:
ETH2:
- IP address: **192.168.3.2**
- Gateway: **192.168.3.1**
- DHCP server:
  - Lease range start: **192.168.3.200**
  - Lease range end: **192.168.3.250**
  - Custom gateway: **192.168.3.3**
- Surelink: **Enabled**
  - Interval: **15s**
  - Test type: **Ping test**
  - Ping host: **my.devicecloud.com**

VRRP:
- Interface: **ETH2**
- Router ID: **50**
- Virtual IP: **192.168.3.3**
- Priority: **80**

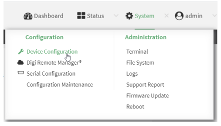
VRRP+
- Monitor interface: **Modem**
- Monitor VRRP+ master: **True**
- Weight: **30**

Configure device one (master device)

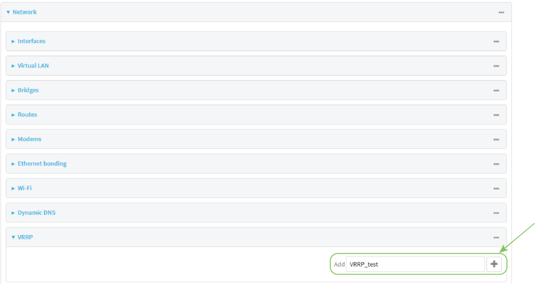
WebUI

Task 1: Configure VRRP on device one

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

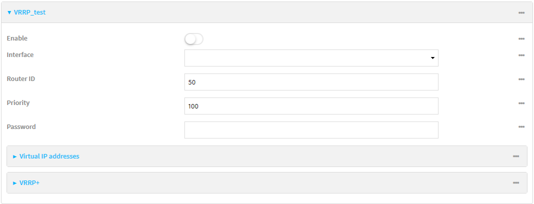
   

   The **Configuration** window is displayed.
3. Click **Network** > **VRRP**.
4. For **Add VRRP instance**, type a name for the VRRP instance and click ✚.

   Add VRRP_test ✚

   The new VRRP instance configuration is displayed.



▾ VRRP_test

Enable

Interface

Router ID 50

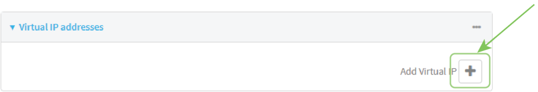
Priority 100

Password

▸ Virtual IP addresses

▸ VRRP+

5. Click **Enable**.
6. For **Interface**, select **Interface: ETH2**.
7. For **Router ID**, leave at the default setting of **50**.
8. For **Priority**, leave at the default setting of **100**.
9. Click to expand **Virtual IP addresses**.
10. Click ✚ to add a virtual IP address.



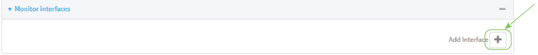
▾ Virtual IP addresses

Add Virtual IP ✚

11. For **Virtual IP**, type **192.168.3.3**.

Task 2: Configure VRRP+ on device one

1. Click to expand **VRRP+**.
2. Click **Enable**.
3. Click to expand **Monitor interfaces**.
4. Click ✚ to add an interface for monitoring.



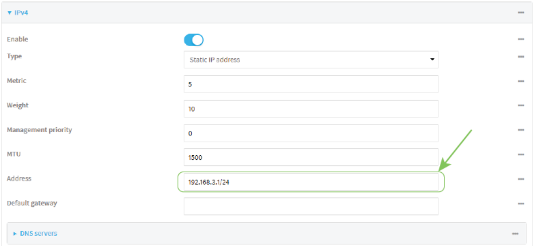
▾ Monitor interfaces

Add Interface ✚

5. Select **Interface: Modem**.
6. For **Priority modifier**, type **30**.

Task 3: Configure the IP address for the VRRP interface, ETH2, on device one

1. Click **Network** > **Interfaces** > **ETH2** > **IPv4**
2. For **Address**, type **192.168.3.1/24**.



▾ IPv4

Enable

Type Static IP address

Metric 5

Weight 10

Management priority 0

MTU 1500

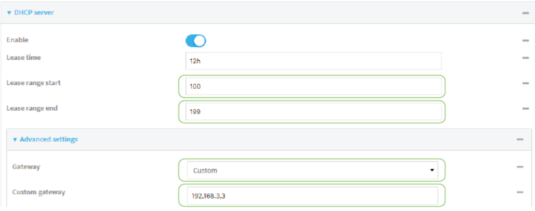
Address 192.168.3.1/24

Default gateway

▸ DNS servers

Task 4: Configure the DHCP server for ETH2 on device one

1. Click to expand **Network** > **Interfaces** > **ETH2** > **IPv4** > **DHCP Server**
2. For **Lease range start**, leave at the default of **100**.
3. For **Lease range end**, type **199**.
4. Click to expand **Advanced settings**.
5. For **Gateway**, select **Custom**.
6. For **Custom gateway**, enter **192.168.3.3**.



DHCP server

Enable: ●

Lease time: 12h

Lease range start: 100

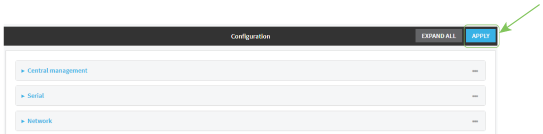
Lease range end: 199

Advanced settings

Gateway: Custom

Custom gateway: 192.168.3.3

7. Click **Apply** to save the configuration and apply the change.



Command line

Task 1: Configure VRRP on device one

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Create the VRRP instance:

```
(config)> add network vrrp VRRP_test
(config network vrrp VRRP_test)>
```

4. Enable the VRRP instance:

```
(config network vrrp VRRP_test)> enable true
(config network vrrp VRRP_test)>
```

5. Set the VRRP interface to ETH2:

```
(config network vrrp VRRP_test)> interface /network/interface/eth2
(config network vrrp VRRP_test)>
```

6. Add the virtual IP address associated with this VRRP instance.

```
(config network vrrp VRRP_test)> add virtual_address end 192.168.3.3
(config network vrrp VRRP_test)>
```

Task 2: Configure VRRP+ on device one

1. Enable VRRP+:

```
(config network vrrp VRRP_test)> vrrp_plus enable true
(config network vrrp VRRP_test )>
```

2. Add the interface to monitor:

```
(config network vrrp VRRP_test)> add vrrp_plus monitor_interface end
/network/interface/modem
(config network vrrp VRRP_test)>
```

3. Set the amount that the device's priority should be decreased or increased due to SureLink connectivity failure or success to **30**:

```
(config network vrrp VRRP_test )> network vrrp VRRP_test vrrp_plus weight
30
(config network vrrp VRRP_test )>
```

Task 3: Configure the IP address for the VRRP interface, ETH2, on device one

1. Type **...** to return to the root of the config prompt:

```
(config network vrrp VRRP_test )> ...
(config)>
```

2. Set the IP address for ETH2:

```
(config)> network interface eth2 ipv4 address 192.168.3.1/24
(config)>
```

Task 4: Configure the DHCP server for ETH2 on device one

1. Set the start and end addresses of the DHCP pool to use to assign DHCP addresses to clients:
   a. Set the start address to **100**:

   ```
   (config)> network interface eth2 ipv4 dhcp_server lease_start 100
   (config)>
   ```

   b. Set the end address to **199**:

   ```
   (config)> network interface eth2 ipv4 dhcp_server lease_end 199
   (config)>
   ```

2. Set the DHCP server gateway type to custom:

```
(config)> network interface eth2 ipv4 dhcp_server advanced gateway custom
(config)>
```

3. Set the custom gateway to **192.168.3.3**:

```
(config)> network interface eth2 ipv4 dhcp_server advanced gateway_custom
192.168.3.3
(config)>
```

4. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

5. Type **exit** to exit the Admin CLI.

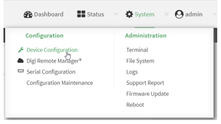
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure device two (backup device)

☰ **WebUI**

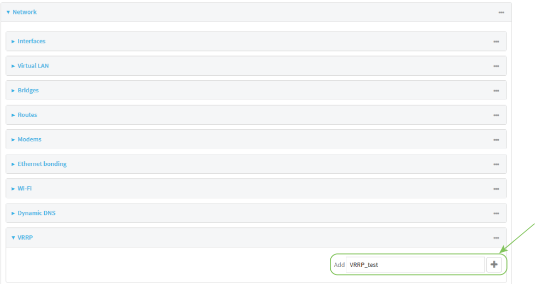
Task 1: Configure VRRP on device two

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



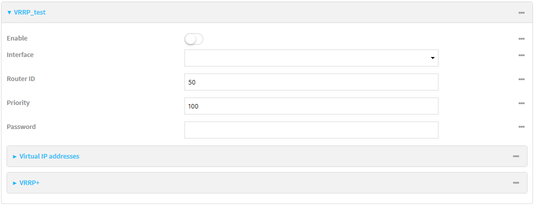
   The **Configuration** window is displayed.

3. Click **Network** > **VRRP**.
4. For **Add VRRP instance**, type a name for the VRRP instance and click **+**.



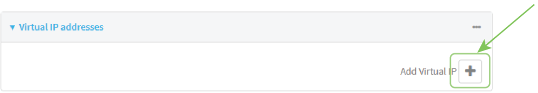
Add VRRP_test

   The new VRRP instance configuration is displayed.



Enable
Interface
Router ID 50
Priority 100
Password

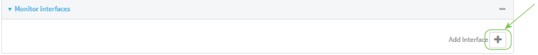
5. Click **Enable**.
6. For **Interface**, select **Interface: ETH2**.
7. For **Router ID**, leave at the default setting of **50**.
8. For **Priority**, type **80**.
9. Click to expand **Virtual IP addresses**.
10. Click ✚ to add a virtual IP address.



11. For **Virtual IP**, type **192.168.3.3**.

Task 2: Configure VRRP+ on device two

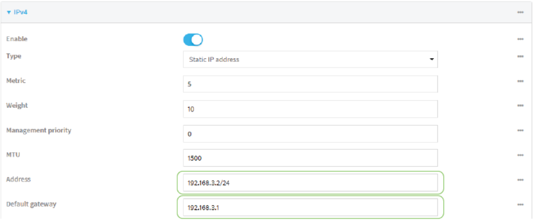
1. Click to expand **VRRP+**.
2. Click **Enable**.
3. Click to expand **Monitor interfaces**.
4. Click ✚ to add an interface for monitoring.



5. Select **Interface: Modem**.
6. Click to enable **Monitor VRRP+ master**.
7. For **Priority modifier**, type **30**.

Task 3: Configure the IP address for the VRRP interface, ETH2, on device two

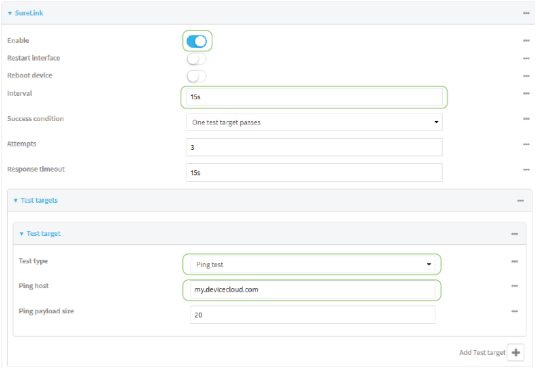
1. Click **Network** > **Interfaces** > **ETH2** > **IPv4**
2. For **Address**, type **192.168.3.2/24**.
3. For **Default gateway**, type the IP address of the VRRP interface on the master device, configured above in Task 3, step 2 (**192.168.3.1**).



Enable
Type Static IP address
Metric 5
Weight 10
Management priority 0
MTU 1500
Address 192.168.3.2/24
Default gateway 192.168.3.1

Task 4: Configure SureLink for ETH2 on device two

1. Click **Network** > **Interfaces** > **ETH2** > **IPv4** > **SureLink**.
2. Click **Enable**.
3. For **Interval**, type **15s**.
4. Click to expand **Test targets** > **Test target**.
5. For **Test Type**, select **Ping test**.
6. For **Ping host**, type **my.devicecloud.com**.



SureLink

Enable: on

Restart interface: off

Reboot device: off

Interval: 15s

Success condition: One test target passes

Attempts: 3

Response timeout: 15s

Test targets

Test target

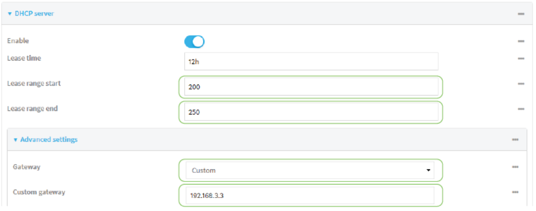
Test type: Ping test

Ping host: my.devicecloud.com

Ping payload size: 20

Task 5: Configure the DHCP server for ETH2 on device two

1. Click to expand **Network** > **Interfaces** > **ETH2** > **IPv4** > **DHCP Server**
2. For **Lease range start**, type **200**.
3. For **Lease range end**, type **250**.
4. Click **Advanced settings**.
5. For **Gateway**, select **Custom**.
6. For **Custom gateway**, enter **192.168.3.3**.



DHCP server

Enable: on

Lease time: 12h

Lease range start: 200

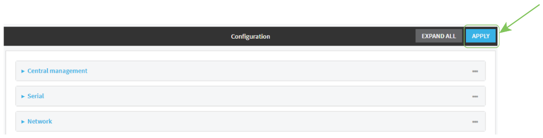
Lease range end: 250

Advanced settings

Gateway: Custom

Custom gateway: 192.168.3.3

7. Click **Apply** to save the configuration and apply the change.



Command line

Task 1: Configure VRRP on device two

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Create the VRRP instance:

   ```
   (config)> add network vrrp VRRP_test
   (config network vrrp VRRP_test)>
   ```
4. Enable the VRRP instance:

   ```
   (config network vrrp VRRP_test)> enable true
   (config network vrrp VRRP_test)>
   ```
5. Set the VRRP interface to ETH2:

   ```
   (config network vrrp VRRP_test)> interface /network/interface/eth2
   (config network vrrp VRRP_test)>
   ```
6. Add the virtual IP address associated with this VRRP instance.

   ```
   (config network vrrp VRRP_test)> add virtual_address end 192.168.3.3
   (config network vrrp VRRP_test)>
   ```

Task 2: Configure VRRP+ on device two

1. Enable VRRP+:

   ```
   (config network vrrp VRRP_test)> vrrp_plus enable true
   (config network vrrp VRRP_test )>
   ```
2. Add the interface to monitor:

   ```
   (config network vrrp VRRP_test)> add vrrp_plus monitor_interface end
   /network/interface/modem
   (config network vrrp VRRP_test)>
   ```
3. Enable the ability to monitor the master device:

   ```
   (config network vrrp VRRP_test)> vrrp_plus monitor_master true
   (config network vrrp VRRP_test)>
   ```
4. Set the amount that the device's priority should be decreased or increased due to SureLink connectivity failure or success to **30**:

   ```
   (config network vrrp VRRP_test )> network vrrp VRRP_test vrrp_plus weight
   30
   (config network vrrp VRRP_test )>
   ```

Task 3: Configure the IP address for the VRRP interface, ETH2, on device two

1. Type **...** to return to the root of the config prompt:

```
(config network vrrp VRRP_test )> ...
(config)>
```

2. Set the IP address for ETH2:

```
(config)> network interface eth2 ipv4 address 192.168.3.2
(config)>
```

3. Set the default gateway to the IP address of the VRRP interface on the master device, configured above in Task 3, step 2 (**192.168.3.1**).

```
(config)> network interface eth2 ipv4 gateway 192.168.3.1
(config)>
```

Task 4: Configure SureLink for ETH2 on device two

1. Enable SureLink on the ETH2 interface:

```
(config)> network interface eth2 ipv4 surelink enable true
(config)>
```

2. Create a SureLink test target:

```
(config)> add network interface eth2 ipv4 surelink target end
(config network interface eth2 ipv4 surelink target 0)>
```

3. Set the type of test to ping:

```
(config network interface eth2 ipv4 surelink target 0)> test ping
(config network interface eth2 ipv4 surelink target 0)>
```

4. Set **my.devicecloud.com** as the hostname to ping:

```
(config network interface eth2 ipv4 surelink target 0)> ping_host
my.devicecloud.com
(config network interface eth2 ipv4 surelink target 0)>
```

Task 5: Configure the DHCP server for ETH2 on device two

1. Type **...** to return to the root of the configuration prompt:

```
(config network interface eth2 ipv4 surelink target 0)> ...
(config)>
```

2. Set the start and end addresses of the DHCP pool to use to assign DHCP addresses to clients:

   a. Set the start address to **200**:

   ```
   (config)> network interface eth2 ipv4 dhcp_server lease_start 200
   (config)>
   ```

b. Set the end address to **250**:

```
(config)> network interface eth2 ipv4 dhcp_server lease_end 250
(config)>
```

3. Set the DHCP server gateway type to custom:

```
(config)> network interface eth2 ipv4 dhcp_server advanced gateway custom
(config)>
```

4. Set the custom gateway to **192.168.3.3**:

```
(config)> network interface eth2 ipv4 dhcp_server advanced gateway_custom
192.168.3.3
(config)>
```

5. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

6. Type **exit** to exit the Admin CLI.

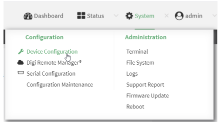
   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show VRRP status and statistics

This section describes how to display VRRP status and statistics for a AnywhereUSB device. VRRP status is available from the Web UI only.

WebUI

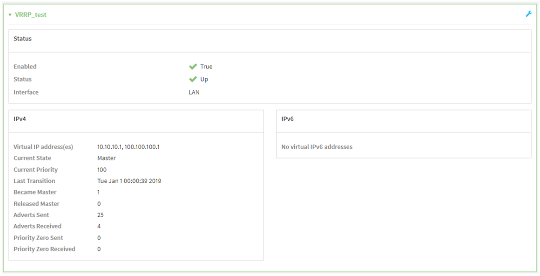
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.

3. Click **Status** > **VRRP**.

   The **Virtual Router Redundancy Protocol** window is displayed.



Status

Enabled	✓ True
Status	✓ Up
Interface	LAN

IPv4

Virtual IP address(es)	10.10.10.1, 100.100.100.1
Current State	Master
Current Priority	100
Last Transition	Tue Jan 1 00:00:39 2019
Became Master	1
Released Master	0
Adverts Sent	25
Adverts Received	4
Priority Zero Sent	0
Priority Zero Received	0

IPv6

No virtual IPv6 addresses

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the Admin CLI prompt, type show vrrp:

```
> show vrrp

 VRRP       Status  Proto  State   Virtual IP
 ----       ------  -----  ------  -------------
 VRRP_test  Up      IPv4   Backup  10.10.10.1
 VRRP_test  Up      IPv4   Backup  100.100.100.1
>
```

3. To display additional information about a specific VRRP instance, at the Admin CLI prompt, type show vrrp name *name*:

```
> show vrrp name VRRP_test

 VRRP_test VRRP Status
 ---------------------
 Enabled                  : True
 Status                   : Up
 Interface                : lan

 IPv4
   ----
   Virtual IP address(es) : 10.10.10.1, 100.100.100.1
   Current State          : Master
   Current Priority       : 100
   Last Transition        : Tue Jan  1 00:00:39 2019
   Became Master          : 1
   Released Master        : 0
   Adverts Sent           : 71
   Adverts Received       : 4
   Priority Zero Sent     : 0
   Priority zero Received : 0

>
```

Virtual Private Networks (VPN)

Virtual Private Networks (VPNs) are used to securely connect two private networks together so that devices can connect from one network to the other using secure channels.

This chapter contains the following topics:

IPsec

IPsec is a suite of protocols for creating a secure communication link—an IPsec tunnel—between a host and a remote IP network or between two IP networks across a public network such as the Internet.

IPsec data protection

IPsec protects the data being sent across a public network by providing the following:

Data origin authentication
Authentication of data to validate the origin of data when it is received.

Data integrity
Authentication of data to ensure it has not been modified during transmission.

Data confidentiality
Encryption of data sent across the IPsec tunnel to ensure that an unauthorized device cannot read the data.

Anti-Replay
Authentication of data to ensure an unauthorized device has not injected it into the IPsec tunnel.

IPsec modes

IPsec can run in two different modes: **Tunnel** and **Transport**.

Tunnel
The entire IP packet is encrypted and/or authenticated and then encapsulated as the payload in a new IP packet.

Transport
Only the payload of the IP packet is encrypted and/or authenticated. The IP header is left untouched. This mode has limitations when using an authentication header, because the IP addresses in the IP header cannot be translated (for example, with Network Address Translation (NAT), as it would invalidate the authentication hash value.

Internet Key Exchange (IKE) settings

IKE is a key management protocol that allows IPsec to negotiate the security associations (SAs) that are used to create the secure IPsec tunnel. Both IKEv1 and IKEv2 are supported.

SA negotiations are performed in two phases, known as **phase 1** and **phase 2**.

Phase 1

In phase 1, IKE creates a secure authenticated communication channel between the device and the peer (the remote device which is at the other end of the IPsec tunnel) using the configured pre-shared key and the Diffie-Hellman key exchange. This creates the IKE SAs that are used to encrypt further IKE communications.

For IKEv1, there are two modes for the phase 1 negotiation: **Main mode** and **Aggressive mode**. IKEv2 does not use these modes.

Main mode
Main mode is the default mode. It is slower than aggressive mode, but more secure, in that all sensitive information sent between the device and its peer is encrypted.

Aggressive mode
Aggressive mode is faster than main mode, but is not as secure as main mode, because the device and its peer exchange their IDs and hash information in clear text instead of being encrypted.

Aggressive mode is usually used when one or both of the devices have a dynamic external IP address.

Phase 2

In phase 2, IKE negotiates the SAs for IPsec. This creates two unidirectional SAs, one for each direction. Once the phase 2 negotiation is complete, the IPsec tunnel should be fully functional.

IPsec and IKE renegotiation

To reduce the chances of an IPsec tunnel being compromised, the IPsec SAs and IKE SA are renegotiated at a regular interval. This results in different encryption keys being used in the IPsec tunnel.

Authentication

Client authenticaton

XAUTH (extended authentication) pre-shared key authentication mode provides additional security by using client authentication credentials in addition to the standard pre-shared key. The AnywhereUSB Plus device can be configured to authenticate with the remote peer as an XAUTH client.

RSA Signatures

With RSA signatures authentication, the AnywhereUSB Plus device uses a private RSA key to authenticate with a remote peer that is using a corresponding public key.

Certificate-based Authentication

X.509 certificate-based authentication makes use of private keys on both the server and client which are secured and never shared. Both the server and client have a certificate which is generated with their respective private key and signed by a Certificate Authority (CA).

The AnywhereUSB Plus implementation of IPsec can be configured to use X.509 certificate-based authentication using the private keys and certificates, along with a root CA certificate from the signing authority and, if available, a Certificate Revocation List (CRL).

Configure an IPsec tunnel

Configuring an IPsec tunnel with a remote device involves configuring the following items:

Required configuration items

- **IPsec tunnel configuration items:**
  - The mode: either tunnel or transport.
  - Enable the IPsec tunnel.

    The IPsec tunnel is enabled by default.
  - The firewall zone of the IPsec tunnel.
  - The routing metric for routes associated with this IPsec tunnel.
  - The authentication type and pre-shared key or other applicable keys and certificates.
  - The local endpoint type and ID values, and the remote endpoint host and ID values.

- **IKE configuration items**
  - The IKE version, either IKEv1 or IKEv2.
  - Whether to initiate a key exchange or wait for an incoming request.
  - The IKE mode, either main aggressive.
  - The IKE authentication protocol to use for the IPsec tunnel negotiation during phase 1 and phase 2.
  - The IKE encryption protocol to use for the IPsec tunnel negotiation during phase 1 and phase 2.
  - The IKE Diffie-Hellman group to use for the IPsec tunnel negotiation during phase 1 and phase 2.
- Enable dead peer detection and configure the delay and timeout.
- Destination networks that require source NAT.
- Active recovery configuration. See Configure SureLink active recovery for IPsec for information about IPsec active recovery.

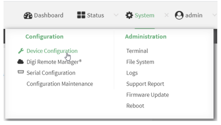
Additional configuration items

The following additional configuration settings are not typically configured to get an IPsec tunnel working, but can be configured as needed:

- Determine whether the device should use UDP encapsulation even when it does not detect that NAT is being used.
- If using IPsec failover, identify the primary tunnel during configuration of the backup tunnel.
- The Network Address Translation (NAT) keep alive time.
- The protocol, either Encapsulating Security Payload (ESP) or Authentication Header (AH).
- The management priority for the IPsec tunnel interface. The active interface with the highest management priority will have its address reported as the preferred contact address for central management and direct device access.
- Enable XAUTH client authentication, and the username and password to be used to authenticate with the remote peer.
- Enable Mode-configuration (MODECFG) to receive configuration information, such as the private IP address, from the remote peer.
- Disable the padding of IKE packets. This should normally not be done except for compatibility purposes.
- Destination networks that require source NAT.
- Depending on your network and firewall configuration, you may need to add a packet filtering rule to allow incoming IPsec traffic.
- **Tunnel and key renegotiating**
  - The lifetime of the IPsec tunnel before it is renegotiated.
  - The amount of time before the IKE phase 1 lifetime expires.
  - The amount of time before the IKE phase 2 lifetime expires
  - The lifetime margin, a randomizing amount of time before the IPsec tunnel is renegotiated.

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

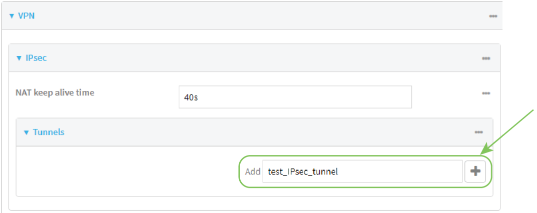
   

   The **Configuration** window is displayed.
3. Click **VPN** > **IPsec**.
4. (Optional) Change the **NAT keep alive time**.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **NAT keep alive time** to ten minutes, enter **10m** or **600s**.

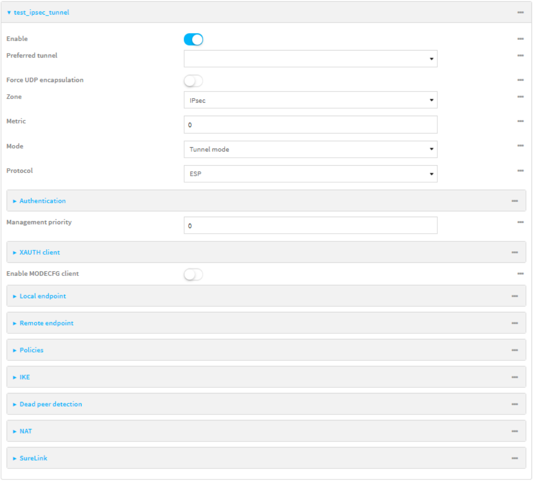
   The default is 40 seconds.
5. Click to expand **Tunnels**.
6. For **Add IPsec tunnel**, type a name for the tunnel and click ✚.

   NAT keep alive time: 40s

   Add: test_IPsec_tunnel ✚

   The new IPsec tunnel configuration is displayed.

   Enable: ●
   Preferred tunnel:
   Force UDP encapsulation: ○
   Zone: IPsec
   Metric: 0
   Mode: Tunnel mode
   Protocol: ESP
   Management priority: 0
   Enable MODECFG client: ○
7. The IPsec tunnel is enabled by default. To disable, click **Enable**.
8. (Optional) **Preferred tunnel** provides an optional mechanism for IPsec failover behavior. See Configure IPsec failover for more information.

9. (Optional) Enable **Force UDP encapsulation** to force the tunnel to use UDP encapsulation even when it does not detect that NAT is being used.
10. For **Zone**, select the firewall zone for the IPsec tunnel. Generally this should be left at the default of **IPsec**.

Note Depending on your network configuration, you may need to add a packet filtering rule to allow incoming traffic. For example, for the **IPsec** zone:

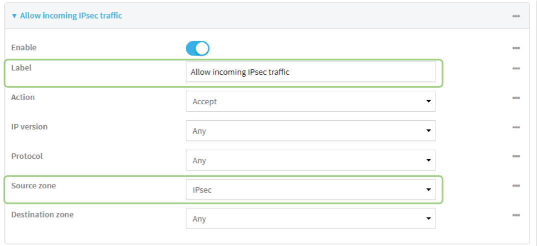
a. Click to expand **Firewall** > **Packet filtering**.

b. For **Add packet filter**, click ✚.

c. For **Label**, type **Allow incoming IPsec traffic**.

d. For Source zone, select **IPsec**.

Leave all other fields at their default settings.



▾ Allow incoming IPsec traffic

Enable: ●

Label: Allow incoming IPsec traffic

Action: Accept

IP version: Any

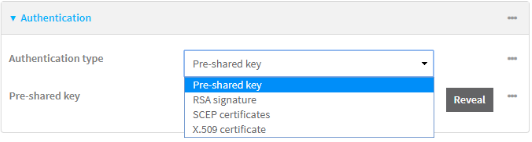
Protocol: Any

Source zone: IPsec

Destination zone: Any

11. For **Metric**, enter or select the priority of routes associated with this IPsec tunnel. When more than one active route matches a destination, the route with the lowest metric is used.

    The metric can also be used in tandem with SureLink to configure IPsec failover behavior. See Configure IPsec failover for more information.
12. Select the Mode, either:
    - **Tunnel**: The entire IP packet is encrypted and/or authenticated and then encapsulated as the payload in a new IP packet.
    - **Transport**: Only the payload of the IP packet is encrypted and/or authenticated. The IP header is unencrypted.
13. Select the **Protocol**, either:
    - **ESP** (Encapsulating Security Payload): Provides encryption as well as authentication and integrity.
    - **AH** (Authentication Header): Provides authentication and integrity only.
14. Click to expand **Authentication**.



▾ Authentication

Authentication type: Pre-shared key

Pre-shared key:

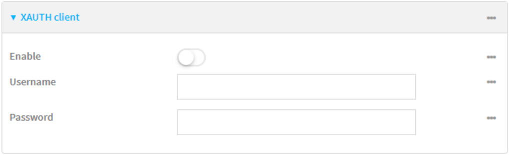
a. For **Authentication type**, select one of the following:

- **Pre-shared key**: Uses a pre-shared key (PSK) to authenticate with the remote peer.

    i. Type the **Pre-shared key**.

- **Asymmetric pre-shared keys**: Uses asymmetric pre-shared keys to authenticate with the remote peer.
    1. For **Local key**, type the local pre-shared key. This must be the same as the remote key on the remote host.
    2. For **Remote key**, type the remote pre-shared key. This must be the same as the local key on the remote host.
- **SA signature**: Uses a private RSA key to authenticate with the remote peer.
    1. For **Private key**, paste the device's private RSA key in PEM format.
    2. Type the **Private key passphrase** that is used to decrypt the private key. Leave blank if the private key is not encrypted.
    3. For **Peer public key**, paste the peer's public RSA key in PEM format.
- **X.509 certificate**: Uses private key and X.509 certificates to authenticate with the remote peer.
    1. For **Private key**, paste the device's private RSA key in PEM format.
    2. Type the **Private key passphrase** that is used to decrypt the private key. Leave blank if the private key is not encrypted.
    3. For **Certificate**, paste the local X.509 certificate in PEM format.
    4. For Peer verification, select either:
        - **Peer certificate**: For **Peer certificate**, paste the peer's X.509 certificate in PEM format.
        - **Certificate Authority**: For **Certificate Authority chain**, paste the Certificate Authority (CA) certificates. These must include all peer certificates in the chain up to the root CA certificate, in PEM format.

15. (Optional) For **Management Priority**, set the priority for this IPsec tunnel.
16. (Optional) To configure the device to connect to its remote peer as an XAUTH client:
    a. Click to expand **XAUTH client**.



    b. Click **Enable**.
    c. Type the **Username** and **Password** that the device will use to authenticate as an XAUTH client with the peer.
17. (Optional) Click **Enable MODECFG client** to receive configuration information, such as the private IP address, from the remote peer.
18. Click to expand **Local endpoint**.
    a. For **Type**, select either:
        - **Default route**: Uses the same network interface as the default route.
        - **Interface**: Select the **Interface** to be used as the local endpoint.

   b. Click to expand **ID**.
      i. Select the ID type:
         - **Auto**: The ID will be automatically determined from the value of the tunnels endpoints.
         - **Raw**: Enter an ID and have it passed unmodified to the underlying IPsec stack.

           For **Raw ID value**, type the ID that will be passed.
         - **Any**: Any ID will be accepted.
         - **IPv4**: The ID will be interpreted as an IP address and sent as an ID_IPV4_ADDR IKE identity.

           For **IPv4 ID value**, type an IPv4 formatted ID. This can be a fully-qualified domain name or an IPv4 address.
         - **IPv6**: The ID will be interpreted as an IP address and sent as an ID_IPV6_ADDR IKE identity.

           For **IPv6 ID value**, type an IPv6 formatted ID. This can be a fully-qualified domain name or an IPv6 address.
         - **RFC822/Email**: The ID will be interpreted as an RFC822 (email address).

           For **RFC822 ID value**, type the ID in internet email address format.
         - **FQDN**: The ID will be interpreted as FQDN (Fully Qualified Domain Name) and sent as an ID_FQDN IKE identity.

           For **FQDN ID value**, type the ID as an FQDN.
         - **KeyID**: The ID will be interpreted as a Key ID and sent as an ID_KEY_ID IKE identity.

           For **KEYID ID value**, type the key ID.

19. Click to expand **Remote endpoint**.
   a. For **Hostname**, select either a hostname or IP address. If your device is not configured to initiate the IPsec connection (see **IKE** > **Initiate connection**), you can also use the keyword **any**, which means that the hostname is dynamic or unknown.
   b. Click to expand **ID**.
      i. Select the ID type:
         - **Auto**: The ID will be automatically determined from the value of the tunnels endpoints.
         - **Raw**: Enter an ID and have it passed unmodified to the underlying IPsec stack.

           For **Raw ID value**, type the ID that will be passed.
         - **Any**: Any ID will be accepted.
         - **IPv4**: The ID will be interpreted as an IPv4 address and sent as an ID_IPV4_ADDR IKE identity.

           For **IPv4 ID value**, type an IPv4 formatted ID. This can be a fully-qualified domain name or an IPv4 address.
         - **IPv6**: The ID will be interpreted as an IPv6 address and sent as an ID_IPV6_ADDR IKE identity.

           For **IPv6 ID value**, type an IPv6 formatted ID. This can be a fully-qualified domain name or an IPv6 address.

- **RFC822/Email**: The ID will be interpreted as an RFC822 (email address).

  For **RFC822 ID value**, type the ID in internet email address format.
- **FQDN**: The ID will be interpreted as FQDN (Fully Qualified Domain Name) and sent as an ID_FQDN IKE identity.

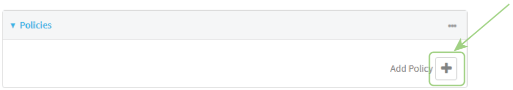
  For **FQDN ID value**, type the ID as an FQDN.
- **KeyID**: The ID will be interpreted as a Key ID and sent as an ID_KEY_ID IKE identity.

  For **KEYID ID value**, type the key ID.

20. Click to expand **Policies**.

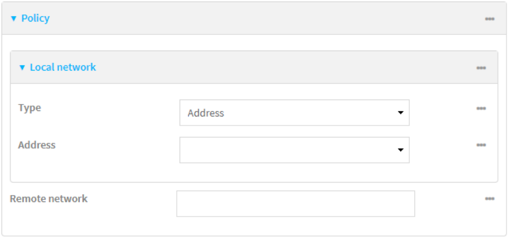
    Policies define the network traffic that will be encapsulated by this tunnel.

    a. Click ✚ to create a new policy.

    The new policy configuration is displayed.

    b. Click to expand **Local network**.

    Type: Address
    Address:
    Remote network:

    c. For **Type**, select one of the following:

    - **Address**: The address of a local network interface.

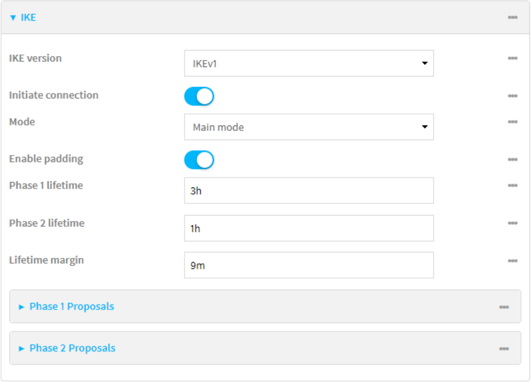
      For **Address**, select the appropriate interface.
    - **Network**: The subnet of a local network interface.

      For **Address**, select the appropriate interface.
    - **Custom network**: A user-defined network.

      For **Custom network**, enter the IPv4 address and optional netmask. The keyword **any** can also be used.
    - **Request a network**: Requests a network from the remote peer.

    d. For **Remote network**, enter the IP address and optional netmask of the remote network. The keyword **any** can also be used. .

21. Click to expand **IKE**.



▾ IKE

IKE version: IKEv1

Initiate connection: ●

Mode: Main mode

Enable padding: ●

Phase 1 lifetime: 3h

Phase 2 lifetime: 1h

Lifetime margin: 9m

▸ Phase 1 Proposals

▸ Phase 2 Proposals

   a. For **IKE version**, select either IKEv1 or IKEv2. This setting must match the peer's IKE version.
   b. **Initiate connection** instructs the device to initiate the key exchange, rather than waiting for an incoming request. This must be disabled if **Remote endpoint** > **Hostname** is set to **any**.
   c. For **Mode**, select either **Main mode** or **Aggressive mode**.
   d. For **Enable padding**, click to disable the padding of IKE packets. This should normally not be disabled except for compatibility purposes.
   e. For Phase 1 lifetime, enter the amount of time that the IKE security association expires after a successful negotiation and must be re-authenticated.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Phase 1 lifetime** to ten minutes, enter **10m** or **600s**.
   f. For Phase 2 lifetime, enter the amount of time that the IKE security association expires after a successful negotiation and must be rekeyed.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Phase 2 lifetime** to ten minutes, enter **10m** or **600s**.
   g. For Lifetime margin, enter a randomizing amount of time before the IPsec tunnel is renegotiated.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Lifetime margin** to ten minutes, enter **10m** or **600s**.
   h. Click to expand **Phase 1 Proposals**.
      i. Click ✚ to create a new phase 1 proposal.
      ii. For **Cipher**, select the type of encryption.
      iii. For **Hash**, select the type of hash to use to verify communication integrity.
      iv. For **Diffie-Hellman group**, select the type of Diffie-Hellman group to use for key exchange.
      v. You can add additional Phase 1 proposals by clicking ✚ next to **Add Phase 1 Proposal**.

i. Click to expand **Phase 2 Proposals**.

   i. Click ✚ to create a new phase 2 proposal.

   ii. For **Cipher**, select the type of encryption.

   iii. For **Hash**, select the type of hash to use to verify communication integrity.

   iv. For **Diffie-Hellman group**, select the type of Diffie-Hellman group to use for key exchange.

   v. You can add additional Phase 2 proposals by clicking ✚ next to **Add Phase 2 Proposal**.

22. (Optional) Click to expand **Dead peer detection**. Dead peer detection is enabled by default. Dead peer detection uses periodic IKE transmissions to the remote endpoint to detect whether tunnel communications have failed, allowing the tunnel to be automatically restarted when failure occurs.

    a. To enable or disable dead peer detection, click **Enable**.

    b. For **Delay**, type the number of seconds between transmissions of dead peer packets. Dead peer packets are only sent when the tunnel is idle.

    c. For **Timeout**, type the number of seconds to wait for a response from a dead peer packet before assuming the tunnel has failed.

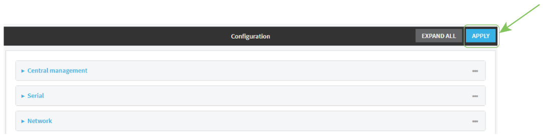
23. (Optional) Click to expand **NAT** to create a list of destination networks that require source NAT.

    a. Click ✚ next to **Add NAT destination**.

    b. For **Destination network**, type the IPv4 address and optional netmask of a destination network that requires source NAT. You can also use **any**, meaning that any destination network connected to the tunnel will use source NAT.

24. See Configure SureLink active recovery for IPsec for information about IPsec **Active recovery**.

25. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add an IPsec tunnel. For example, to add an IPsec tunnel named **ipsec_example**:

```
(config)> add vpn ipsec tunnel ipsec_example
(config vpn ipsec tunnel ipsec_example)>
```

   The IPsec tunnel is enabled by default. To disable:

```
(config vpn ipsec tunnel ipsec_example)> enable false
(config vpn ipsec tunnel ipsec_example)>
```

4. (Optional) Set the tunnel to use UDP encapsulation even when it does not detect that NAT is being used:

```
(config vpn ipsec tunnel ipsec_example)> force_udp_encap true
(config vpn ipsec tunnel ipsec_example)>
```

5. Set the firewall zone for the IPsec tunnel. Generally this should be left at the default of **ipsec**.

```
(config vpn ipsec tunnel ipsec_example)> zone zone
(config vpn ipsec tunnel ipsec_example)>
```

   To view a list of available zones:

```
(config vpn ipsec tunnel ipsec_example)> zone ?

Zone: The firewall zone assigned to this IPsec tunnel. This can be used by
packet filtering rules
and access control lists to restrict network traffic on this tunnel.
Format:
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup
Default value: ipsec
Current value: ipsec

(config vpn ipsec tunnel ipsec_example)>
```

Note Depending on your network configuration, you may need to add a packet filtering rule to allow incoming traffic. For example, for the **IPsec** zone:

a. Type **...** to move to the root of the configuration:

```
(config vpn ipsec tunnel ipsec_example)> ...
(config)>
```

b. Add a packet filter:

```
(config)> add firewall filter end
(config firewall filter 2)>
```

c. Set the label to **Allow incoming IPsec traffic**:

```
(config config firewall filter 2)> label "Allow incoming IPsec traffic"
(config firewall filter 2)>
```

d. Set the source zone to **ipsec**:

```
(config config firewall filter 2)> src_zone ipsec
(config firewall filter 2)>
```

6. Set the metric for the IPsec tunnel. When more than one active route matches a destination, the route with the lowest metric is used. The metric can also be used in tandem with SureLink to configure IPsec failover behavior. See Configure IPsec failover for more information.

```
(config vpn ipsec tunnel ipsec_example)> metric value
(config vpn ipsec tunnel ipsec_example)>
```

where *value* is any integer between **0** and **65535**.

7. Set the mode:

```
(config vpn ipsec tunnel ipsec_example)> mode mode
(config vpn ipsec tunnel ipsec_example)>
```

where *mode* is either:

- **tunnel**: The entire IP packet is encrypted and/or authenticated and then encapsulated as the payload in a new IP packet.
- **transport**: Only the payload of the IP packet is encrypted and/or authenticated. The IP header is unencrypted.

The default is **tunnel**.

8. Set the protocol:

```
(config vpn ipsec tunnel ipsec_example)> type protocol
(config vpn ipsec tunnel ipsec_example)>
```

where *protocol* is either:

- **esp** (Encapsulating Security Payload): Provides encryption as well as authentication and integrity.
- **ah** (Authentication Header): Provides authentication and integrity only.

The default is **esp**.

9. (Optional) Set the management priority for this IPsec tunnel:

   ```
   (config vpn ipsec tunnel ipsec_example)> mgmt value
   (config vpn ipsec tunnel ipsec_example)>
   ```

   where *value* is any interger between **0** and **1000**.

10. Set the authentication type:

    ```
    (config vpn ipsec tunnel ipsec_example)> auth type value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is one of:

    - **secret**: Uses a pre-shared key (PSK) to authenticate with the remote peer.
      a. Set the pre-shared key:

      ```
      (config vpn ipsec tunnel ipsec_example)> auth secret key
      (config vpn ipsec tunnel ipsec_example)>
      ```

    - **asymmetric-secrets**: Uses asymmetric pre-shared keys to authenticate with the remote peer.
      a. Set the local pre-shared key. This must be the same as the remote key on the remote host.:

      ```
      (config vpn ipsec tunnel ipsec_example)> auth local_secret key
      (config vpn ipsec tunnel ipsec_example)>
      ```

      b. Set the remote pre-shared key. This must be the same as the local key on the remote host.:

      ```
      (config vpn ipsec tunnel ipsec_example)> auth remote_secret key
      (config vpn ipsec tunnel ipsec_example)>
      ```

    - **rsasig**: Uses a private RSA key to authenticate with the remote peer.
      a. For the **private_key** parameter, paste the device's private RSA key in PEM format:

      ```
      (config vpn ipsec tunnel ipsec_example)> auth private_key key
      (config vpn ipsec tunnel ipsec_example)>
      ```

      b. Set the private key passphrase that is used to decrypt the private key. Leave blank if the private key is not encrypted.

      ```
      (config vpn ipsec tunnel ipsec_example)> auth private_key_
      passphrase passphrase
      (config vpn ipsec tunnel ipsec_example)>
      ```

      c. For the **peer_public_key** parameter, paste the peer's public RSA key in PEM format:

      ```
      (config vpn ipsec tunnel ipsec_example)> auth peer_public_key key
      (config vpn ipsec tunnel ipsec_example)>
      ```

- **x509**: Uses private key and X.509 certificates to authenticate with the remote peer.

    a. For the **private_key** parameter, paste the device's private RSA key in PEM format:

    ```
    (config vpn ipsec tunnel ipsec_example)> auth private_key key
    (config vpn ipsec tunnel ipsec_example)>
    ```

    b. Set the private key passphrase that is used to decrypt the private key. Leave blank if the private key is not encrypted.

    ```
    (config vpn ipsec tunnel ipsec_example)> auth private_key_
    passphrase passphrase
    (config vpn ipsec tunnel ipsec_example)>
    ```

    c. For the **cert** parameter, paste the local X.509 certificate in PEM format:

    ```
    (config vpn ipsec tunnel ipsec_example)> auth cert certificate
    (config vpn ipsec tunnel ipsec_example)>
    ```

    d. Set the method for verifying the peer's X.509 certificate:

    ```
    (config vpn ipsec tunnel ipsec_example)> auth peer_verify value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is either:

    - **cert**: Uses the peer's X.509 certificate in PEM format for verification.
        - For the **peer_cert** parameter, paste the peer's X.509 certificate in PEM format:

        ```
        (config vpn ipsec tunnel ipsec_example)> auth peer_cert
        certificate
        (config vpn ipsec tunnel ipsec_example)>
        ```

    - **ca**: Uses the Certificate Authority chain for verification.
        - For the **ca_cert** parameter, paste the Certificate Authority (CA) certificates. These must include all peer certificates in the chain up to the root CA certificate, in PEM format.

        ```
        (config vpn ipsec tunnel ipsec_example)> auth ca_cert cert_
        chain
        (config vpn ipsec tunnel ipsec_example)>
        ```

11. (Optional) Configure the device to connect to its remote peer as an XAUTH client:

    a. Enable XAUTH client functionality:

    ```
    (config vpn ipsec tunnel ipsec_example)> xauth_client enable true
    (config vpn ipsec tunnel ipsec_example)>
    ```

    b. Set the XAUTH client username:

    ```
    (config vpn ipsec tunnel ipsec_example)> xauth_client username name
    (config vpn ipsec tunnel ipsec_example)>
    ```

c. Set the XAUTH client password:

```
(config vpn ipsec tunnel ipsec_example)> xauth_client password pwd
(config vpn ipsec tunnel ipsec_example)>
```

12. (Optional) Enable MODECFG client functionality:

    MODECFG client functionality configures the device to receive configuration information, such as the private IP address, from the remote peer.

    a. Enable MODECFG client functionality:

    ```
    (config vpn ipsec tunnel ipsec_example)> modecfg_client enable true
    (config vpn ipsec tunnel ipsec_example)>
    ```

13. Configure the local endpoint:

    a. Set the method for determining the local network interface:

    ```
    (config vpn ipsec tunnel ipsec_example)> local type value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is either:

    - **defaultroute**: Uses the same network interface as the default route.
    - **interface**: Select the **Interface** to be used as the local endpoint.

    b. Set the ID type:

    ```
    (config vpn ipsec tunnel ipsec_example)> local id type value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is one of:

    - **auto**: The ID will be automatically determined from the value of the tunnels endpoints.
    - **raw**: Enter an ID and have it passed unmodified to the underlying IPsec stack.

      Set the unmodified ID that will be passed:

      ```
      (config vpn ipsec tunnel ipsec_example)> local id raw_id id
      (config vpn ipsec tunnel ipsec_example)>
      ```

    - **any**: Any ID will be accepted.
    - **ipv4**: The ID will be interpreted as an IPv4 address and sent as an ID_IPV4_ADDR IKE identity.

      Set an IPv4 formatted ID. This can be a fully-qualified domain name or an IPv4 address.

      ```
      (config vpn ipsec tunnel ipsec_example)> local id ipv4_id id
      (config vpn ipsec tunnel ipsec_example)>
      ```

    - **ipv6**: The ID will be interpreted as an IPv6 address and sent as an ID_IPV6_ADDR IKE identity.

      Set an IPv6 formatted ID. This can be a fully-qualified domain name or an IPv6 address.

```
(config vpn ipsec tunnel ipsec_example)> local id ipv6_id id
(config vpn ipsec tunnel ipsec_example)>
```

- **rfc822**: The ID will be interpreted as an RFC822 (email address).

  Set the ID in internet email address format:

  ```
  (config vpn ipsec tunnel ipsec_example)> local id rfc822_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

- **fqdn**: The ID will be interpreted as FQDN (Fully Qualified Domain Name) and sent as an ID_FQDN IKE identity.

  Set the ID as an FQDN:

  ```
  (config vpn ipsec tunnel ipsec_example)> local id rfc822_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

- **keyid**: The ID will be interpreted as a Key ID and sent as an ID_KEY_ID IKE identity.

  Set the key ID:

  ```
  (config vpn ipsec tunnel ipsec_example)> local id keyid_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

14. Configure the remote endpoint:

    a. Set the hostname or IP address of the remote endpoint:

    ```
    (config vpn ipsec tunnel ipsec_example)> remote hostname value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    If your device is not configured to initiate the IPsec connection (see ike initiate), you can also use the keyword **any**, which means that the hostname is dynamic or unknown.

    b. Set the ID type:

    ```
    (config vpn ipsec tunnel ipsec_example)> remote id type value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is one of:

    - **auto**: The ID will be automatically determined from the value of the tunnels endpoints.
    - **raw**: Enter an ID and have it passed unmodified to the underlying IPsec stack.

      Set the unmodified ID that will be passed:

      ```
      (config vpn ipsec tunnel ipsec_example)> remote id raw_id id
      (config vpn ipsec tunnel ipsec_example)>
      ```

    - **any**: Any ID will be accepted.
    - **ipv4**: The ID will be interpreted as an IPv4 address and sent as an ID_IPV4_ADDR IKE identity.

      Set an IPv4 formatted ID. This can be a fully-qualified domain name or an IPv4 address.

```
(config vpn ipsec tunnel ipsec_example)> remote id ipv4_id id
(config vpn ipsec tunnel ipsec_example)>
```

- **ipv6**: The ID will be interpreted as an IPv6 address and sent as an ID_IPV6_ADDR IKE identity.

  Set an IPv6 formatted ID. This can be a fully-qualified domain name or an IPv6 address.

  ```
  (config vpn ipsec tunnel ipsec_example)> remote id ipv6_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

- **rfc822**: The ID will be interpreted as an RFC822 (email address).

  Set the ID in internet email address format:

  ```
  (config vpn ipsec tunnel ipsec_example)> remote id rfc822_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

- **fqdn**: The ID will be interpreted as FQDN (Fully Qualified Domain Name) and sent as an ID_FQDN IKE identity.

  Set the ID as an FQDN:

  ```
  (config vpn ipsec tunnel ipsec_example)> remote id rfc822_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

- **keyid**: The ID will be interpreted as a Key ID and sent as an ID_KEY_ID IKE identity.

  Set the key ID:

  ```
  (config vpn ipsec tunnel ipsec_example)> remote id keyid_id id
  (config vpn ipsec tunnel ipsec_example)>
  ```

15. Configure IKE settings:

    a. Set the IKE version:

    ```
    (config vpn ipsec tunnel ipsec_example)> ike version value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is either **ikev1** or **ikev2**. This setting must match the peer's IKE version.

    b. Determine whether the device should initiate the key exchange, rather than waiting for an incoming request. By default, the device will initiate the key exchange. This must be disabled if remote hostname is set to **any**. To disable:

    ```
    (config vpn ipsec tunnel ipsec_example)> ike initiate false
    (config vpn ipsec tunnel ipsec_example)>
    ```

    c. Set the IKE phase 1 mode:

    ```
    (config vpn ipsec tunnel ipsec_example)> ike mode value
    (config vpn ipsec tunnel ipsec_example)>
    ```

    where *value* is either **aggressive** or **main**.

d. Padding of IKE packets is enabled by default and should normally not be disabled except for compatibility purposes. To disable:

```
(config vpn ipsec tunnel ipsec_example)> ike pad false
(config vpn ipsec tunnel ipsec_example)>
```

e. Set the amount of time that the IKE security association expires after a successful negotiation and must be re-authenticated:

```
(config vpn ipsec tunnel ipsec_example)> ike phase1_lifetime value
(config vpn ipsec tunnel ipsec_example)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **phase1_lifetime** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example)> ike phase1_lifetime 600s
(config vpn ipsec tunnel ipsec_example)>
```

The default is three hours.

f. Set the amount of time that the IKE security association expires after a successful negotiation and must be rekeyed.

```
(config vpn ipsec tunnel ipsec_example)> ike phase2_lifetime value
(config vpn ipsec tunnel ipsec_example)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **phase2_lifetime** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example)> ike phase2_lifetime 600s
(config vpn ipsec tunnel ipsec_example)>
```

The default is one hour.

g. Set a randomizing amount of time before the IPsec tunnel is renegotiated:

```
(config vpn ipsec tunnel ipsec_example)> ike lifetime_margin value
(config vpn ipsec tunnel ipsec_example)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

For example, to set **lifetime_margin** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example)> ike lifetime_margin 600s
(config vpn ipsec tunnel ipsec_example)>
```

The default is nine minutes.

h. Configure the types of encryption, hash, and Diffie-Hellman group to use during phase 1:

   i. Add a phase 1 proposal:

   ```
   (config vpn ipsec tunnel ipsec_example)> add ike phase1_proposal end
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
   ```

   ii. Set the type of encryption to use during phase 1:

   ```
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
   cipher value
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
   ```

   where *value* is one of **3des**, **aes128**, **aes192**, **aes256**, or **null**. The default is **3des**.

   iii. Set the type of hash to use during phase 1 to verify communication integrity:

   ```
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)> hash
   value
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
   ```

   where *value* is one of **md5**, **sha1**, **sha256**, **sha384**, or **sha512**. The default is **sha1**.

   iv. Set the type of Diffie-Hellman group to use for key exchange during phase 1:

   ```
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)> dh_
   group value
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
   ```

   where *value* is one of **ecp384**, **modp768**, **modp1024**, **modp1536**, **modp2048**, **modp3072**, **modp4096**, **modp6144**, or **modp8192**, . The default is **modp1024**.

   v. (Optional) Add additional phase 1 proposals:

      i. Move back one level in the schema:

      ```
      (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)>
      ..
      (config vpn ipsec tunnel ipsec_example ike phase1_proposal)>
      ```

      ii. Add an additional proposal:

      ```
      (config vpn ipsec tunnel ipsec_example ike phase1_proposal)> add
      end
      (config vpn ipsec tunnel ipsec_example ike phase1_proposal 1)>
      ```

      Repeat the above steps to set the type of encryption, hash, and Diffie-Hellman group for the additional proposal.

      iii. Repeat to add more phase 1 proposals.

i. Configure the types of encryption, hash, and Diffie-Hellman group to use during phase 2:

   i. Move back two levels in the schema:

   ```
   (config vpn ipsec tunnel ipsec_example ike phase1_proposal 0)> .. ..
   (config vpn ipsec tunnel ipsec_example ike)>
   ```

ii. Add a phase 2 proposal:

```
(config vpn ipsec tunnel ipsec_example ike)> add ike phase2_proposal
end
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
```

iii. Set the type of encryption to use during phase 2:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
cipher value
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
```

where *value* is one of **3des**, **aes128**, **aes192**, **aes256**, or **null**. The default is **3des**.

iv. Set the type of hash to use during phase 2 to verify communication integrity:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)> hash
value
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
```

where *value* is one of **md5**, **sha1**, **sha256**, **sha384**, or **sha512**. The default is **sha1**.

v. Set the type of Diffie-Hellman group to use for key exchange during phase 2:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)> dh_
group value
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
```

where *value* is one of **ecp384**, **modp768**, **modp1024**, **modp1536**, **modp2048**, **modp3072**, **modp4096**, **modp6144**, or **modp8192**, . The default is **modp1024**.

vi. (Optional) Add additional phase 2 proposals:

i. Move back one level in the schema:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)>
..
(config vpn ipsec tunnel ipsec_example ike phase2_proposal)>
```

ii. Add an additional proposal:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal)> add
end
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 1)>
```

Repeat the above steps to set the type of encryption, hash, and Diffie-Hellman group for the additional proposal.

iii. Repeat to add more phase 2 proposals.

16. (Optional) Configure dead peer detection:

Dead peer detection is enabled by default. Dead peer detection uses periodic IKE transmissions to the remote endpoint to detect whether tunnel communications have failed, allowing the tunnel to be automatically restarted when failure occurs.

a. Change to the root of the configuration schema:

```
(config vpn ipsec tunnel ipsec_example ike phase2_proposal 0)> ...
(config)>
```

b. To disable dead peer detection:

```
(config)> vpn ipsec tunnel ipsec_example dpd enable false
(config)>
```

c. Set the number of seconds between transmissions of dead peer packets. Dead peer packets are only sent when the tunnel is idle. The default is **60**.

```
(config)> vpn ipsec tunnel ipsec_example dpd delay value
(config)>
```

d. Set the number of seconds to wait for a response from a dead peer packet before assuming the tunnel has failed. The default is **90**.

```
(config)> vpn ipsec tunnel ipsec_example dpd timeout value
(config)>
```

17. (Optional) Create a list of destination networks that require source NAT:

a. Add a destination network:

```
(config)> add vpn ipsec tunnel ipsec_example nat end
(config vpn ipsec tunnel ipsec_example nat 0)>
```

b. Set the IPv4 address and optional netmask of a destination network that requires source NAT. You can also use **any**, meaning that any destination network connected to the tunnel will use source NAT.

```
(config vpn ipsec tunnel ipsec_example nat 0)> dst value
(config vpn ipsec tunnel ipsec_example nat 0)>
```

18. Configure policies that define the network traffic that will be encapsulated by this tunnel:

a. Change to the root of the configuration schema:

```
(config vpn ipsec tunnel ipsec_example nat 0)> ...
(config)>
```

b. Add a policy:

```
(config)> add vpn ipsec tunnel ipsec_example policy end
(config vpn ipsec tunnel ipsec_example policy 0)>
```

c. Set the type of local network policy:

```
(config vpn ipsec tunnel ipsec_example policy 0)> local type value
(config vpn ipsec tunnel ipsec_example policy 0)>
```

where *value* is one of:

- **address**: The address of a local network interface.

  Set the address:

  i. Use the **?** to determine available interfaces:

  ```
  (config vpn ipsec tunnel ipsec_example policy 0)> local
  address ?

  Address: The local network interface to use the address of.
  This field must be set when 'Type' is set to 'Address'.
  Format:
    defaultip
    defaultlinklocal
    eth1
    eth2
    loopback
  Current value:

  (config vpn ipsec tunnel ipsec_example policy 0)> local
  address
  ```

  ii. Set the interface. For example:

  ```
  (config vpn ipsec tunnel ipsec_example policy 0)> local
  address eth1
  (config vpn ipsec tunnel ipsec_example policy 0)>
  ```

- **network**: The subnet of a local network interface.

  Set the network:

  i. Use the **?** to determine available interfaces:

  ```
  (config vpn ipsec tunnel ipsec_example policy 0)> local
  network ?

  Interface: The network interface.
  Format:
    defaultip
    defaultlinklocal
    eth1
    eth2
    loopback
  Current value:

  (config vpn ipsec tunnel ipsec_example policy 0)> local
  network
  ```

ii. Set the interface. For example:

```
(config vpn ipsec tunnel ipsec_example policy 0)> local
network eth1
(config vpn ipsec tunnel ipsec_example policy 0)>
```

- **custom**: A user-defined network.

  Set the custom network:

  ```
  (config vpn ipsec tunnel ipsec_example policy 0)> local custom
  value
  (config vpn ipsec tunnel ipsec_example policy 0)>
  ```

  where *value* is the IPv4 address and optional netmask. The keyword **any** can also be used.

- **request**: Requests a network from the remote peer.

d. Set the IP address and optional netmask of the remote network. The keyword **any** can also be used.

```
(config vpn ipsec tunnel ipsec_example policy 0)> remote network value
(config vpn ipsec tunnel ipsec_example policy 0)>
```

19. (Optional) Change the NAT keep alive time:

a. Change to the root of the configuration schema:

```
(config vpn ipsec tunnel ipsec_example policy 0)> ...
(config)>
```

b.

```
(config)> vpn ipsec advanced keep_alive value
(config)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

For example, to set **keep_alive** to ten minutes, enter either **10m** or **600s**:

```
(config)> vpn ipsec advanced keep_alive 600s
(config)>
```

The default is 40 seconds.

20. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

21. Type **exit** to exit the Admin CLI.

Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure IPsec failover

There are two methods to configure the AnywhereUSB Plus device to fail over from a primary IPsec tunnel to a backup tunnel:

- **SureLink** active recovery—You can use SureLink along with the IPsec tunnel's metric to configure two or more tunnels so that when the primary tunnel is determined to be inactive by SureLink, a secondary tunnel can begin serving traffic that the primary tunnel was serving.
- **Preferred tunnel**—When multiple IPsec tunnels are configured, one tunnel can be configured as a backup to another tunnel by defining a preferred tunnel for the backup device.

Required configuration items

- Two or more configured IPsec tunnels: The primary tunnel, and one or more backup tunnels.
- Either:
  - SureLink configured on the primary tunnel with **Restart Interface** enabled, and the metric for all tunnels set appropriately to determine which IPsec tunnel has priority. With this failover configuration, both tunnels are active simultaneously, and there is minimal downtime due to failover.
  - Identify the preferred tunnel during configuration of the backup tunnel. In this scenario, the backup tunnel is not active until the preferred tunnel fails.

IPsec failover using SureLink

With this configuration, when two IPsec tunnels are configured with the same local and remote endpoints but different metrics, traffic addressed to the remote endpoint will be routed through the IPsec tunnel with the lower metric.

If **SureLink** > **Restart Interface** is enabled for the tunnel with the lower metric, and SureLink determines that the tunnel is not functioning properly (for example, pings to a host at the other end of the tunnel are failing), then:

1. SureLink will shut down the tunnel and renegotiate its IPsec connection.
2. While the tunnel with the lower metric is down, traffic addressed to the remote endpoint will be routed through the tunnel with the higher metric.

For example:

- Tunnel_1:
  - **Metric**: 10
  - **Local endpoint** > **Interface**: ETH2
  - **Remote endpoint** > **Hostname**: 192.168.10.1
  - **SureLink** configuration:
    - **Restart Interface** enabled
    - **Test target**:
      - **Test type**: Ping test
      - **Ping host**: 192.168.10.2
- Tunnel_2:

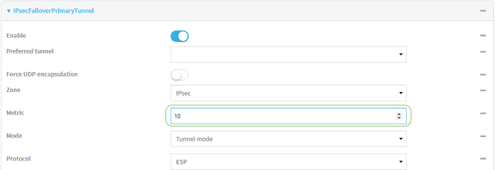
- **Metric**: 20
- **Local endpoint** > **Interface**: ETH2
- **Remote endpoint** > **Hostname**: 192.168.10.1

In this configuration:

1. Tunnel_1 will normally be used for traffic destined for the 192.168.10.1 endpoint.
2. If pings to 192.168.10.2 fail, SureLink will shut down the tunnel and renegotiate its IPsec connection.
3. While Tunnel_1 is down, Tunnel_2 will be used for traffic destined for the 192.168.10.1 endpoint.

WebUI

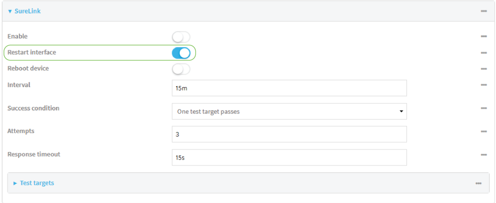
1. Configure the primary IPsec tunnel. See Configure an IPsec tunnel for instructions.
   - During configuration of the IPsec tunnel, set the metric to a low value (for example, **10**).



▾ IPsecFailoverPrimaryTunnel

Enable	●
Preferred tunnel	
Force UDP encapsulation	○
Zone	IPsec
Metric	10
Mode	Tunnel mode
Protocol	ESP

   - Configure SureLink for the primary IPsec tunnel and enable **Restart interface**. See Configure SureLink active recovery for IPsec for instructions.

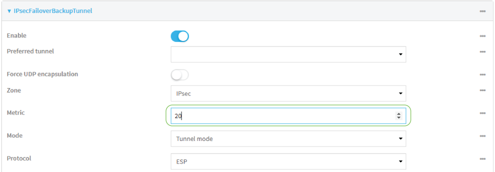


▾ SureLink

Enable	○
Restart interface	●
Reboot device	○
Interval	15m
Success condition	One test target passes
Attempts	3
Response timeout	15s

▸ Test targets

2. Create a backup IPsec tunnel. Configure this tunnel to use the same local and remote endpoints as the primary tunnel. See Configure an IPsec tunnel for instructions.
   - During configuration of the IPsec tunnel, set the metric to a value that is higher than the metric of the primary tunnel (for example, **20**).



▾ IPsecFailoverBackupTunnel

Enable	●
Preferred tunnel	
Force UDP encapsulation	○
Zone	IPsec
Metric	20
Mode	Tunnel mode
Protocol	ESP

Command line

1. Configure the primary IPsec tunnel. See Configure an IPsec tunnel for instructions.
   - During configuration of the IPsec tunnel, set the metric to a low value (for example, **10**):

     ```
     (config vpn ipsec tunnel IPsecFailoverPrimaryTunnel)> metric 10
     (config vpn ipsec tunnel IPsecFailoverPrimaryTunnel)>
     ```

   - Configure SureLink for the primary IPsec tunnel and enable **Restart interface**. See Configure SureLink active recovery for IPsec for instructions.

     ```
     (config vpn ipsec tunnel IPsecFailoverPrimaryTunnel)> surelink restart
     true
     (config vpn ipsec tunnel IPsecFailoverPrimaryTunnel)>
     ```

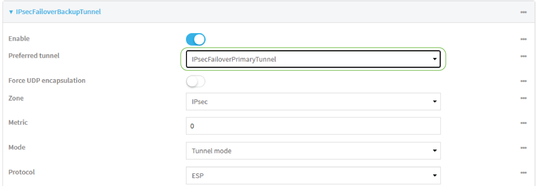
2. Create a backup IPsec tunnel. Configure this tunnel to use the same local and remote endpoints as the primary tunnel. See Configure an IPsec tunnel for instructions.
   - During configuration of the IPsec tunnel, set the metric to a value that is higher than the metric of the primary tunnel (for example, **20**):

     ```
     (config vpn ipsec tunnel IPsecFailoverBackupTunnel)> metric 20
     (config vpn ipsec tunnel IPsecFailoverBackupTunnel)>
     ```

IPsec failover using Preferred tunnel

WebUI

1. Configure the primary IPsec tunnel. See Configure an IPsec tunnel for instructions.
2. Create a backup IPsec tunnel. See Configure an IPsec tunnel for instructions.
3. During configuration of the backup IPsec tunnel, identify the primary IPsec tunnel in the **Preferred tunnel** parameter:



▾ IPsecFailoverBackupTunnel

Enable: ●

Preferred tunnel: IPsecFailoverPrimaryTunnel

Force UDP encapsulation: ○

Zone: IPsec

Metric: 0

Mode: Tunnel mode

Protocol: ESP

Command line

1. Configure the primary IPsec tunnel. See Configure an IPsec tunnel for instructions.
2. Create a backup IPsec tunnel. See Configure an IPsec tunnel for instructions.
3. During configuration of the backup IPsec tunnel, identify the primary IPsec tunnel:
   a. Use the **?** to view a list of available tunnels:

   ```
   (config vpn ipsec tunnel backup_ipsec_tunnel)> ipsec_failover ?

   Preferred tunnel: This tunnel will not start until the preferred tunnel
   has failed. It will continue
   to operate until the preferred tunnel returns to full operation status.
   ```

```
Format:
 primary_ipsec_tunnel
 backup_ipsec_tunnel
Optional: yes
Current value:

(config vpn ipsec tunnel backup_ipsec_tunnel)> ipsec_failover
```

b. Set the primary IPsec tunnel:

```
(config vpn ipsec tunnel backup_ipsec_tunnel)> ipsec_failover primary_
ipsec_tunnel
(config vpn ipsec tunnel backup_ipsec_tunnel)>
```

Configure SureLink active recovery for IPsec

You can configure the AnywhereUSB Plus device to regularly probe IPsec client connections to determine if the connection has failed and take remedial action.

You can also configure the IPsec tunnel to fail over to a backup tunnel. See Configure IPsec failover for further information.

Required configuration items

- A valid IPsec configuration. See Configure an IPsec tunnel for configuration instructions.
- Enable IPsec active recovery.
- The behavior of the AnywhereUSB Plus device upon IPsec failure: either
  - Restart the IPsec interface
  - Reboot the device.

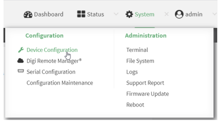
Additional configuration items

- The interval between connectivity tests.
- Whether the interface should be considered to have failed if one of the test targets fails, or all of the test targets fail.
- The number of probe attempts before the IPsec connection is considered to have failed.
- The amount of time that the device should wait for a response to a probe attempt before considering it to have failed.

To configure the AnywhereUSB Plus device to regularly probe the IPsec connection:

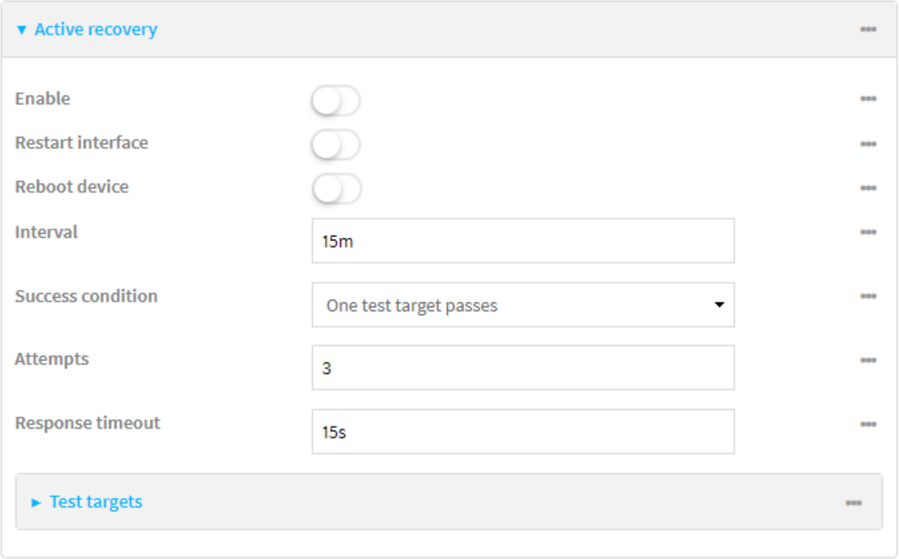
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



The **Configuration** window is displayed.

3. Click **VPN** > **IPsec**.
4. Create a new IPsec tunnel or select an existing one:
   - To create a new IPsec tunnel, see Configure an IPsec tunnel.
   - To edit an existing IPsec tunnel, click to expand the appropriate tunnel.
5. After creating or selecting the IPsec tunnel, click **Active recovery**.



▼ Active recovery

Enable ○

Restart interface ○

Reboot device ○

Interval: 15m

Success condition: One test target passes

Attempts: 3

Response timeout: 15s

► Test targets

6. **Enable** active recovery.
7. For **Restart interface**, enable to configure the device to restart the interface when its connection is considered to have failed. This is useful for interfaces that may regain connectivity after restarting, such as a cellular modem.
8. For **Reboot device**, enable to instruct the device to reboot when the WAN connection is considered to have failed.
9. Change the **Interval** between connectivity tests.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **Interval** to ten minutes, enter **10m** or **600s**.

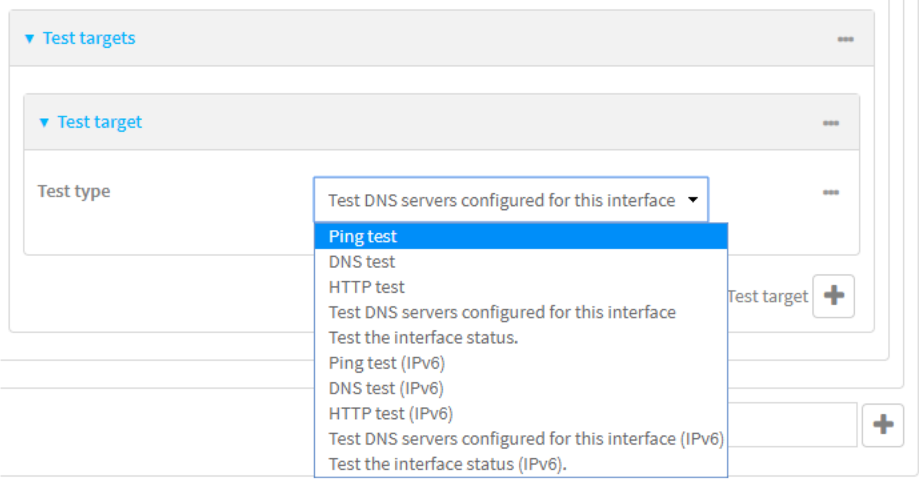
   The default is 15 minutes.
10. For **Success condition**, determine whether the interface should fail over based on the failure of one of the test targets, or all of the test targets.
11. For **Attempts**, type the number of probe attempts before the WAN is considered to have failed.
12. For **Response timeout**, type the amount of time that the device should wait for a response to a probe attempt before considering it to have failed.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **Response timeout** to ten minutes, enter **10m** or **600s**.

    The default is 15 seconds.

13. Add a test target:
    a. Click to expand **Test targets**.



Test type: Test DNS servers configured for this interface

Test targets
Test target
Ping test
DNS test
HTTP test
Test DNS servers configured for this interface
Test the interface status.
Ping test (IPv6)
DNS test (IPv6)
HTTP test (IPv6)
Test DNS servers configured for this interface (IPv6)
Test the interface status (IPv6).
Test target

    b. For **Add Test target**, click ✚.
    c. Select the **Test type**:

- **Ping test** or **Ping test (IPv6)**: Tests connectivity by sending an ICMP echo request to the hostname or IP address specified in **Ping host**. You can also optionally change the number of bytes in the **Ping payload size**.
- **DNS test** or **DNS test (IPv6)**: Tests connectivity by sending a DNS query to the specified **DNS server**.
- **HTTP test HTTP test (IPv6)**: Tests connectivity by sending an HTTP or HTTPS GET request to the URL specified in **Web servers**. The URL should take the format of **http[s]://*hostname*/[*path*]**.
- **Test DNS servers configured for this interface** or **Test DNS servers configured for this interface (IPv6)**: Tests connectivity by sending a DNS query to the DNS servers configured for this interface.
- **Test the interface status** or **Test the interface status IPv6**: The interface is considered to be down based on:
    - **Down time**: The amount of time that the interface can be down before this test is considered to have failed.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Down time** to ten minutes, enter **10m** or **600s**.

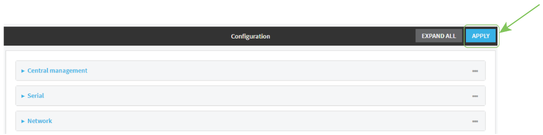
      The default is 60 seconds.
    - **Initial connection time**: The amount of time to wait for an initial connection to the interface before this test is considered to have failed.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Initial connection time** to ten minutes, enter **10m** or **600s**.

      The default is 60 seconds.

14. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Create a new IPsec tunnel, or edit an existing one:
   - To create a new IPsec tunnel, see Configure an IPsec tunnel.
   - To edit an existing IPsec tunnel, change to the IPsec tunnel's node in the configuration schema. For example, for an IPsec tunnel named **ipsec_example**, change to the **ipsec_example** node in the configuration schema:

```
(config)> vpn ipsec tunnel ipsec_example
(config vpn ipsec tunnel ipsec_example)>
```

4. Enable active recovery:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor enable true
(config vpn ipsec tunnel ipsec_example)>
```

5. To configure the device to restart the interface when its connection is considered to have failed:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor restart true
(config vpn ipsec tunnel ipsec_example)>
```

   This is useful for interfaces that may regain connectivity after restarting, such as a cellular modem.
6. To configure the device to reboot when the interface is considered to have failed:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor reboot enable
(config vpn ipsec tunnel ipsec_example)>
```

7. Set the **Interval** between connectivity tests:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor interval value
(config vpn ipsec tunnel ipsec_example)>
```

   where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number*{w|d|h|m|s}**.

For example, to set **interval** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor interval 600s
(config vpn ipsec tunnel ipsec_example)>
```

The default is 15 minutes.

8. Determine whether the interface should fail over based on the failure of one of the test targets, or all of the test targets:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor success_
condition value
(config vpn ipsec tunnel ipsec_example)>
```

Where *value* is either **one** or **all**.

9. Set the number of probe attempts before the WAN is considered to have failed:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor attempts num
(config vpn ipsec tunnel ipsec_example)>
```

The default is **3**.

10. Set the amount of time that the device should wait for a response to a probe attempt before considering it to have failed:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor timeout value
(config vpn ipsec tunnel ipsec_example)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

For example, to set **timeout** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example)> connection_monitor interval 600s
(config vpn ipsec tunnel ipsec_example)>
```

The default is 15 seconds.

11. Configure test targets:

a. Add a test target:

```
(config vpn ipsec tunnel ipsec_example)> add connection_monitor target
end
(config vpn ipsec tunnel ipsec_example connection_monitor target 0)>
```

b. Set the test type:

```
(config vpn ipsec tunnel ipsec_example connection_monitor target 0)>
test value
(config vpn ipsec tunnel ipsec_example connection_monitor target 0)>
```

where *value* is one of:

- **ping** (IPv4) or **ping6** (IPv6): Tests connectivity by sending an ICMP echo request to a specified hostname or IP address.

- Specify the hostname or IP address by using **ping_host** or **ping_host6**:

  ```
  (config vpn ipsec tunnel ipsec_example connection_monitor
  target 0)> ping_host host
  (config vpn ipsec tunnel ipsec_example connection_monitor
  target 0)>
  ```

- (Optional) Set the size, in bytes, of the ping packet by using **ping_size** or **ping_size6**:

  ```
  (config vpn ipsec tunnel ipsec_example connection_monitor
  target 0)> ping_size [num]
  (config vpn ipsec tunnel ipsec_example connection_monitor
  target 0)>
  ```

- **dns** (IPv4) or **dns6** (IPv6): Tests connectivity by sending a DNS query to the specified DNS server.
  - Specify the DNS server. Allowed value is the IP address of the DNS server.

    ```
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)> dns_server ip_address
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)>
    ```

- **dns_configured** (IPv4) or **dns_configured6** (IPv6): Tests connectivity by sending a DNS query to the DNS servers configured for this interface.
- **http** (IPv4) or **http6** (IPv6): Tests connectivity by sending an HTTP or HTTPS GET request to the specified URL.
  - Specify the url. Allowed value uses the format **http[s]://*hostname*/[*path*]**.

    ```
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)> http_url url
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)>
    ```

- **interface_up** (IPv4) or **interface_up6** (IPv6): : The interface is considered to be down based on the interfaces down time, and the amount of time an initial connection to the interface takes before this test is considered to have failed.
  - (Optional) Set the amount of time that the interface can be down before this test is considered to have failed:

    ```
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)> interface_down_time value
    (config vpn ipsec tunnel ipsec_example connection_monitor
    target 0)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **interface_down_time** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)> interface_down_time 600s
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)>
```

The default is 60 seconds.

- (Optional) Set the amount of time to wait for an initial connection to the interface before this test is considered to have failed:

```
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)> interface_timeout value
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)>
```

where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

For example, to set **interface_timeout** to ten minutes, enter either **10m** or **600s**:

```
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)> interface_timeout 600s
(config vpn ipsec tunnel ipsec_example connection_monitor
target 0)>
```

The default is 60 seconds.

12. Save the configuration and apply the change:

```
(config vpn ipsec tunnel ipsec_example connection_monitor target 0)> save
Configuration saved.
>
```

13. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show IPsec status and statistics

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, select **Status** > **IPsec**.

   The **IPsec** page appears.
3. To view configuration details about an IPsec tunnel, click the 🔧 (configuration) icon in the upper right of the tunnel's status pane.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. To display details about all configured IPsec tunnels, type the following at the prompt:

```
> show ipsec all

 Name    Enable  Status   Hostname
 ------  ------  -------  ---------------
 ipsec1  true    up       192.168.2.1
 vpn1    false   pending  192.168.3.1

>
```

3. To display details about a specific tunnel:

```
> show ipsec tunnel ipsec1

 Tunnel                    : ipsec1
 Enable                    : true
 Status                    : pending
 Hostname                  : 192.168.2.1
 Zone                      : ipsec
 Mode                      : tunnel
 Type                      : esp

>
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Debug an IPsec configuration

If you experience issues with an IPsec tunnel not being successfully negotiated with the remote end of the tunnel, you can enable IPsec debug messages to be written to the system log. See View system and event logs for more information about viewing the system log.

There are two methods to enable IPsec debug messages:

- From the Admin CLI—Sets the debug level to **1** (basic debugging information only).
- From the interactive shell—Allows for more detailed debug information.

Use the Admin CLI to set the IPsec debug level to 1

To set the debug level to **1** by using the Admin CLI:

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Set the **action ipsec debug** command to **true**:

   ```
   config> action ipsec debug true
   config>
   ```

4. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

5. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

This sets the IPsec debug level to **1**.

Use the interactive shell to set the IPsec debug level

By using the interactive shell to set the debug level, you can enable the AnywhereUSB Plus device to write additional debug messages to the system log. The command accepts the following values to set the debug level:

- **-1** — (Default) No debug information is written. This is the equivalent of turning off debug messages for IPsec.
- **0** — Basic auditing logs, (for example, SA up/SA down).
- **1** — Generic control flow with errors. Select this for basic debugging information.
- **2** — More detailed debugging control flow.
- **3** — Includes RAW data dumps in hexadecimal format.
- **4** — Also includes sensitive material in dumps (for example, encryption keys).

To access the shell menu option, you must have shell access enabled. See Authentication groups for information about configuring authentication groups that include shell access.

Command line

1. Log into the AnywhereUSB Plus command line as a user with shell access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **shell** to access the device shell.

2. At the shell prompt, execute the following command:

   ```
   # ipsec stroke loglevel ike debug_level
   #
   ```

   where *debug_level* is one of the following:

   - **-1** — (Default) No debug information is written. This is the equivalent of turning off debug messages for IPsec.

- **0** — Basic auditing logs, (for example, SA up/SA down).
- **1** — Generic control flow with errors. Select this for basic debugging information.
- **2** — More detailed debugging control flow.
- **3** — Includes RAW data dumps in hexadecimal format.
- **4** — Also includes sensitive material in dumps (for example, encryption keys).

3. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

OpenVPN

OpenVPN is an open-source Virtual Private Network (VPN) technology that creates secure point-to-point or site-to-site connections in routed or bridged configurations. OpenVPN uses a custom security protocol that is Secure Socket Layer (SSL) / Transport Layer Security (TLS) for key exchange. It uses standard encryption and authentication algorithms for data privacy and authentication over TCP or UDP.

The OpenVPN server can push the network configuration, such as the topology and IP routes, to OpenVPN clients. This makes OpenVPN simpler to configure as it reduces the chances of a configuration mismatch between the client and server. OpenVPN also supports cipher negotiation between the client and server. This means you can configure the OpenVPN server and clients with a range of different cipher options and the server will negotiate with the client on the cipher to use for the connection.

Note The AnywhereUSB can only be configured as an OpenVPN client. It cannot be configured as an OpenVPN server.

For more information on OpenVPN, see www.openvpn.net.

OpenVPN modes:

There are two modes for running OpenVPN:

- Routing mode, also known as TUN.
- Bridging mode, also known as TAP.

Routing (TUN) mode

In routing mode, each OpenVPN client is assigned a different IP subnet from the OpenVPN server and other OpenVPN clients. OpenVPN clients use Network Address Translation (NAT) to route traffic from devices connected on its LAN interfaces to the OpenVPN server.

The manner in which the IP subnets are defined depends on the OpenVPN topology in use. The AnywhereUSB Plus device supports two types of OpenVPN topology:

OpenVPN Topology	Subnet definition method
net30	Each OpenVPN client is assigned a **/30** subnet within the IP subnet specified in the OpenVPN server configuration. With net30 topology, pushed routes are used, with the exception of the default route. . Automatic route pushing (exec) is not allowed, because this would not inform the firewall and would be blocked.
subnet	Each OpenVPN client connected to the OpenVPN server is assigned an IP address within the IP subnet specified in the OpenVPN server configuration. For the AnywhereUSB Plus device, pushed routes are not allowed; you will need to manually configure routes on the device.

For more information on OpenVPN topologies, see OpenVPN topology.

Bridging (TAP) mode

In bridging mode, a LAN interface on the OpenVPN server is assigned to OpenVPN. The LAN interfaces of the OpenVPN clients are on the same IP subnet as the OpenVPN server's LAN interface. This means that devices connected to the OpenVPN client's LAN interface are on the same IP subnet as devices. The AnywhereUSB Plus device supports two mechanisms for configuring an OpenVPN server in TAP mode:

- OpenVPN managed—The AnywhereUSB Plus device creates the interface and then uses its standard configuration to set up the connection (for example, its standard DHCP server configuration).
- Device only—IP addressing is controlled by the system, not by OpenVPN.

Additional OpenVPN information

For more information on OpenVPN, see these resources:

Bridging vs. routing

OpenVPN/Routing

Configure an OpenVPN server

Required configuration items

- Enable the OpenVPN server.

  The OpenVPN server is enabled by default.
- The mode used by the OpenVPN server, one of:
  - **TUN (OpenVPN managed)**—Also known as routing mode. Each OpenVPN client is assigned a different IP subnet from the OpenVPN server and other OpenVPN clients. OpenVPN clients use Network Address Translation (NAT) to route traffic from devices connected on its LAN interfaces to the OpenVPN server.
  - **TAP - OpenVPN managed**—Also know as bridging mode. A more advanced implementation of OpenVPN. The AnywhereUSB Plus device creates an OpenVPN interface and uses standard interface configuration (for example, a standard DHCP server configuration).
  - **TAP - Device only**—An alternate form of OpenVPN bridging mode, in which the device, rather than OpenVPN, controls the interface configuration. If this method is is, the OpenVPN server must be included as a device in either an interface or a bridge.
- The firewall zone to be used by the OpenVPN server.
- The IP network and subnet mask of the OpenVPN server.
- The server's Certificate authority (CA) certificate, and public, private and Diffie-Hellman (DH) keys.
- An OpenVPN authentication group and an OpenVPN user.
- Determine the method of certificate management:
  - Certificates managed by the server.
  - Certificates created externally and added to the server.
- If certificates are created and added to the server, determine the level of authentication:
  - Certificate authentication only.
  - Username and password authentication only.
  - Certificate and username and password authentication.

If username and password authentication is used, you must create an OpenVPN authentication group and user. See Configure an OpenVPN Authentication Group and User for instructions.

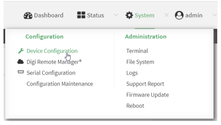
- Certificates and keys:
  - The **CA certificate** (usually in a ca.crt file).
  - The **Public key** (for example, server.crt)
  - The **Private key** (for example, server.key).
  - The **Diffie Hellman key** (usually in dh2048.pem).
- Active recovery configuration. See Configure SureLink active recovery for OpenVPN for information about OpenVPN active recovery.

Additional configuration items

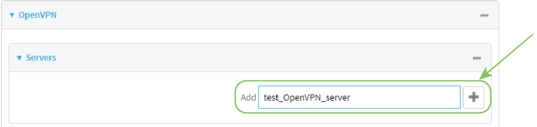
- The route metric for the OpenVPN server.
- The range of IP addresses that the OpenVPN server will provide to clients.
- The TCP/UDP port to use. By default, the AnywhereUSB Plus device uses port **1194**.
- Access control list configuration to restrict access to the OpenVPN server through the firewall.
- Additional OpenVPN parameters.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

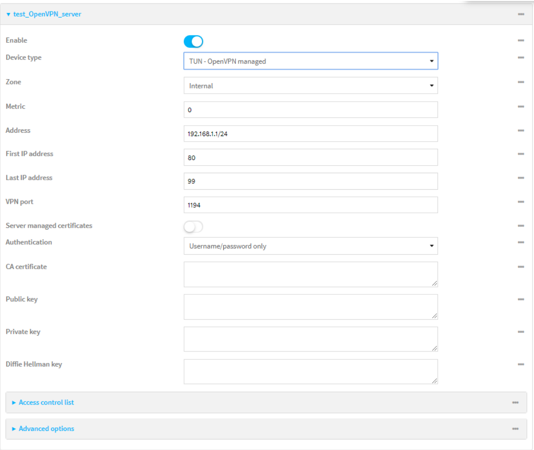
   

   The **Configuration** window is displayed.
3. Click **VPN** > **OpenVPN** > **Servers**.
4. For **Add**, type a name for the OpenVPN server and click ✚.

   Add test_OpenVPN_server

   The new OpenVPN server configuration is displayed.



Enable	●
Device type	TUN - OpenVPN managed
Zone	Internal
Metric	0
Address	192.168.1.1/24
First IP address	80
Last IP address	99
VPN port	1194
Server managed certificates	○
Authentication	Username/password only
CA certificate	
Public key	
Private key	
Diffie Hellman key	

The OpenVPN server is enabled by default. To disable, click **Enable**.

5. For **Device type**, select the mode used by the OpenVPN server, either:
    - **TUN (OpenVPN managed)**
    - **TAP - OpenVPN managed**
    - **TAP - Device only**

    See OpenVPN for information about OpenVPN server modes.
6. If **TUN (OpenVPN managed)** or **TAP - OpenVPN managed** is selected for **Device type**:
    a. For **Zone**, select the firewall zone for the OpenVPN server. For TUN device types, this should be set to **Internal** to treat clients as LAN devices.

    b. (Optional) Select the **Metric** for the OpenVPN server. If multiple active routes match a destination, the route with the lowest metric will be used. The default setting is **0**.

    c. For **Address**, type the IP address and subnet mask of the OpenVPN server.

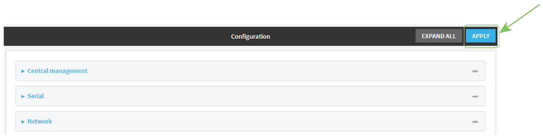
    d. (Optional) For **First IP address** and **Last IP address**, set the range of IP addresses that the OpenVPN server will use when providing IP addresses to clients. The default is from **80** to **99**.
7. (Optional) Set the **VPN port** that the OpenVPN server will use. The default is **1194**.
8. For **Server managed certificates**, determine the method of certificate management. If enabled, the server will manage certificates. If not enabled, certificates must be created externally and added to the server.
9. If **Server managed certificates** is not enabled:

    a. Select the **Authentication** type:
    - **Certificate only**: Uses only certificates for client authentication. Each client requires a public and private key.
    - **Username/password only**: Uses a username and password for client authentication. You must create an OpenVPN authentication group and user. See Configure an OpenVPN Authentication Group and User for instructions.
    - **Certificate and username/password**: Uses both certificates and a username and password for client authentication. Each client requires a public and private key, and you must create an OpenVPN authentication group and user. See Configure an OpenVPN Authentication Group and User for instructions.

   b. Paste the contents of the **CA certificate** (usually in a ca.crt file), the **Public key** (for example, server.crt), the **Private key** (for example, server.key), and the **Diffie Hellman key** (usually in dh2048.pem) into their respective fields. The contents will be hidden when the configuration is saved.

10. (Optional) Click to expand **Access control list** to restrict access to the OpenVPN server:
    - To limit access to specified IPv4 addresses and networks:
      a. Click **IPv4 Addresses**.
      b. For **Add Address**, click ✚.
      c. For **Address**, enter the IPv4 address or network that can access the device's service-type. Allowed values are:
         - A single IP address or host name.
         - A network designation in CIDR notation, for example, 192.168.1.0/24.
         - **any**: No limit to IPv4 addresses that can access the service-type.
      d. Click ✚ again to list additional IP addresses or networks.
    - To limit access to specified IPv6 addresses and networks:
      a. Click **IPv6 Addresses**.
      b. For **Add Address**, click ✚.
      c. For **Address**, enter the IPv6 address or network that can access the device's service-type. Allowed values are:
         - A single IP address or host name.
         - A network designation in CIDR notation, for example, 2001:db8::/48.
         - **any**: No limit to IPv6 addresses that can access the service-type.
      d. Click ✚ again to list additional IP addresses or networks.
    - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:
      a. Click **Interfaces**.
      b. For **Add Interface**, click ✚.
      c. For **Interface**, select the appropriate interface from the dropdown.
      d. Click ✚ again to allow access through additional interfaces.
    - To limit access based on firewall zones:
      a. Click **Zones**.
      b. For **Add Zone**, click ✚.
      c. For **Zone**, select the appropriate firewall zone from the dropdown.

         See Firewall configuration for information about firewall zones.
      d. Click ✚ again to allow access through additional firewall zones.
11. (Optional) Click to expand **Advanced Options** to manually set additional OpenVPN parameters.
    a. Click **Enable** to enable the use of additional OpenVPN parameters.
    b. Click **Override** if the additional OpenVPN parameters should override default options.
    c. For **OpenVPN parameters**, type the additional OpenVPN parameters.
12. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. At the config prompt, type:

```
(config)> add vpn openvpn server name
(config vpn openvpn server name)>
```

   where *name* is the name of the OpenVPN server.

   The OpenVPN server is enabled by default. To disable the server, type:

```
(config vpn openvpn server name)> enable false
(config vpn openvpn server name)>
```

4. Set the mode used by the OpenVPN server:

```
(config vpn openvpn server name)> device_type value
(config vpn openvpn server name)>
```

   where *value* is one of:

   - **TUN (OpenVPN managed)**—Also known as routing mode. Each OpenVPN client is assigned a different IP subnet from the OpenVPN server and other OpenVPN clients. OpenVPN clients use Network Address Translation (NAT) to route traffic from devices connected on its LAN interfaces to the OpenVPN server.
   - **TAP - OpenVPN managed**—Also know as bridging mode. A more advanced implementation of OpenVPN. The AnywhereUSB Plus device creates an OpenVPN interface and uses standard interface configuration (for example, a standard DHCP server configuration).
   - **TAP - Device only**—An alternate form of OpenVPN bridging mode, in which the device, rather than OpenVPN, controls the interface configuration. If this method is is, the OpenVPN server must be included as a device in either an interface or a bridge.

   See OpenVPN for information about OpenVPN modes. The default is **tun**.

5. If **tap** or **tun** are set for **device_type**:

   a. Set the IP address and subnet mask of the OpenVPN server.

   ```
   (config vpn openvpn server name)> address ip_address/netmask
   (config vpn openvpn server name)>
   ```

   b. Set the firewall zone for the OpenVPN server. For TUN device types, this should be set to **internal** to treat clients as LAN devices.

   ```
   (config vpn openvpn server name)> zone value
   (config vpn openvpn server name)>
   ```

   To view a list of available zones:

   ```
   (config vpn openvpn server name)> firewall zone ?

   Zone: The zone for the local TUN interface. To treat clients as LAN
   devices this would usually be
   set to internal.
   Format:
     any
     dynamic_routes
     edge
     external
     internal
     ipsec
     loopback
     setup
   Current value:

   (config vpn openvpn server name)>
   ```

   c. (Optional) Set the route metric for the OpenVPN server. If multiple active routes match a destination, the route with the lowest metric will be used.

   ```
   (config vpn openvpn server name)> metric value
   (config vpn openvpn server name)>
   ```

   where *value* is an interger between **0** and **65535**. The default is **0**.

   d. (Optional) Set the range of IP addresses that the OpenVPN server will use when providing IP addresses to clients:

      i. Set the first address in the range limit:

      ```
      (config vpn openvpn server name)> server_first_ip value
      (config vpn openvpn server name)>
      ```

      where *value* is a number between **1** and **255**. The number entered here will represent the first client IP address. For example, if **address** is set to **192.168.1.1/24** and **server_first_ip** is set to **80**, the first client IP address will be 192.168.1.80.

      The default is from **80**.

ii. Set the last address in the range limit:

```
(config vpn openvpn server name)> server_last_ip value
(config vpn openvpn server name)>
```

where *value* is a number between **1** and **255**. The number entered here will represent the last client IP address. For example, if **address** is set to **192.168.1.1/24** and **server_last_ip** is set to **99**, the last client IP address will be 192.168.1.80.

The default is from **80**.

6. (Optional) Set the port that the OpenVPN server will use:

```
(config vpn openvpn server name)> port port
(config vpn openvpn server name)>
```

The default is **1194**.

7. Determine the method of certificate management:

a. To allow the server to manage certificates:

```
(config vpn openvpn server name)> autogenerate true
(config vpn openvpn server name)>
```

b. To create certificates externally and add them to the server

```
(config vpn openvpn server name)> autogenerate false
(config vpn openvpn server name)>
```

The default setting is **false**.

c. If **autogenerate** is set to false:

i. Set the authentication type:

```
(config vpn openvpn server name)> authentication value
(config vpn openvpn server name)>
```

where *value* is one of:

- **cert**: Uses only certificates for client authentication. Each client requires a public and private key.
- **passwd**: Uses a username and password for client authentication. You must create an OpenVPN authentication group and user. See Configure an OpenVPN Authentication Group and User for instructions.
- **cert_passwd**: Uses both certificates and a username and password for client authentication. Each client requires a public and private key, and you must create an OpenVPN authentication group and user. See Configure an OpenVPN Authentication Group and User for instructions.

ii. Paste the contents of the CA certificate (usually in a ca.crt file) into the value of the **cacert** parameter:

```
(config vpn openvpn server name)> cacert value
(config vpn openvpn server name)>
```

iii. Paste the contents of the public key (for example, server.crt) into the value of the **server_cert** parameter:

```
(config vpn openvpn server name)> server_cert value
(config vpn openvpn server name)>
```

iv. Paste the contents of the private key (for example, server.key) into the value of the **server_key** parameter:

```
(config vpn openvpn server name)> server_key value
(config vpn openvpn server name)>
```

v. Paste the contents of the Diffie Hellman key (usually in dh2048.pem) into the value of the **diffie** parameter:

```
(config vpn openvpn server name)> diffie value
(config vpn openvpn server name)>
```

8. (Optional) Set the access control list to restrict access to the OpenVPN server:

   - To limit access to specified IPv4 addresses and networks:

     ```
     (config vpn openvpn server name)> add acl address end value
     (config vpn openvpn server name)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 192.168.1.0/24.
     - **any**: No limit to IPv4 addresses that can access the service-type.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to specified IPv6 addresses and networks:

     ```
     (config vpn openvpn server name)> add acl address6 end value
     (config vpn openvpn server name)>
     ```

     Where *value* can be:

     - A single IP address or host name.
     - A network designation in CIDR notation, for example, 2001:db8::/48.
     - **any**: No limit to IPv6 addresses that can access the service-type.

     Repeat this step to list additional IP addresses or networks.

   - To limit access to hosts connected through a specified interface on the AnywhereUSB Plus device:

     ```
     (config vpn openvpn server name)> add acl interface end value
     (config vpn openvpn server name)>
     ```

     Where *value* is an interface defined on your device.

     Display a list of available interfaces:

Use **... network interface ?** to display interface information:

```
(config vpn openvpn server name)> ... network interface ?

Interfaces

Additional Configuration
-------------------------------------------
 defaultip               Default IP
 defaultlinklocal        Default Link-local IP
 eth1                    ETH1
 eth2                    ETH2
 loopback                Loopback
 modem                   Modem

(config vpn openvpn server name)>
```

Repeat this step to list additional interfaces.

- To limit access based on firewall zones:

```
(config vpn openvpn server name)> add acl zone end value
```

Where *value* is a firewall zone defined on your device, or the **any** keyword.

Display a list of available firewall zones:

Type **... firewall zone ?** at the config prompt:

```
(config vpn openvpn server name)> ... firewall zone ?

Zones: A list of groups of network interfaces that can be
referred to by packet
filtering rules and access control lists.

 Additional Configuration
 ---------------------------------------------------------
----------------------
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup

(config vpn openvpn server name)>
```

Repeat this step to list additional firewall zones.

9. (Optional) Set additional OpenVPN parameters.

a. Enable the use of additional OpenVPN parameters:

```
(config vpn openvpn server name)> advanced_options enable true
(config vpn openvpn server name)>
```

b. Configure whether the additional OpenVPN parameters should override default options:

```
(config vpn openvpn server name)> advanced_options override true
(config vpn openvpn server name)>
```

c. Set the additional OpenVPN parameters:

```
(config vpn openvpn server name)> extra parameters
(config vpn openvpn server name)>
```

10. Save the configuration and apply the change:

    ```
    (config)> save
    Configuration saved.
    >
    ```

11. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

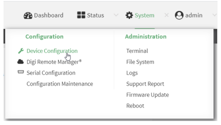
Configure an OpenVPN Authentication Group and User

If username and password authentication is used for the OpenVPN server, you must create an OpenVPN authentication group and user.

See Configure an OpenVPN server for information about configuring an OpenVPN server to use username and password authentication. See AnywhereUSB Plus user authentication for more information about creating authentication groups and users.

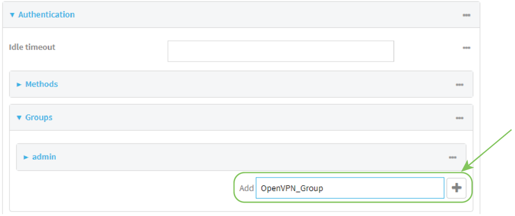
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

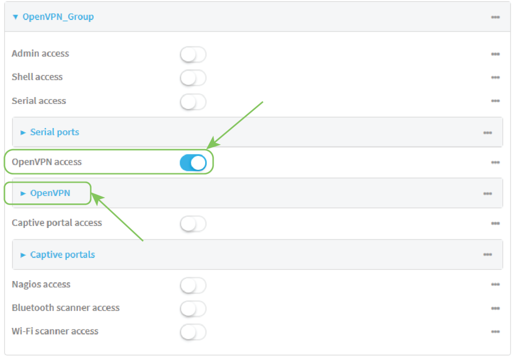
   The **Configuration** window is displayed.

3. Add an OpenVPN authentication group:
   a. Click **Authentication** > **Groups**.
   b. For **Add Group**, type a name for the group (for example, **OpenVPN_Group**) and click ✚.



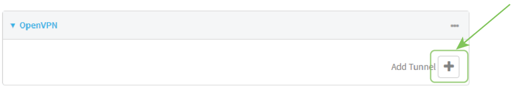
Add OpenVPN_Group ✚

The new authentication group configuration is displayed.

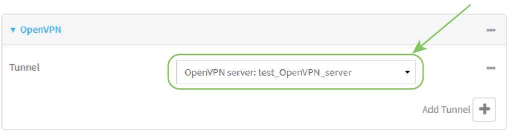


OpenVPN access ●

   c. Click **OpenVPN access** to enable OpenVPN access rights for users of this group.
   d. Click to expand the **OpenVPN** node.
   e. Click ✚ to add a tunnel.



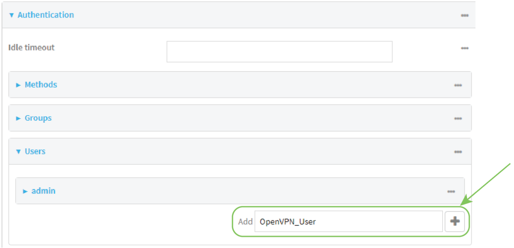
   f. For **Tunnel**, select an OpenVPN tunnel to which users of this group will have access.



Tunnel: OpenVPN server: test_OpenVPN_server

   g. Repeat to add additional OpenVPN tunnels.

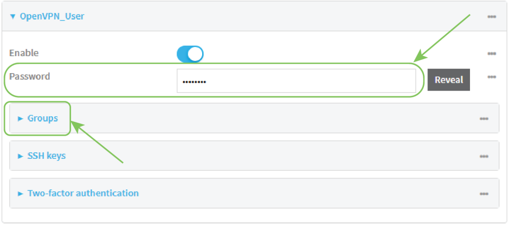
4. Add an OpenVPN authentication user:
   a. Click **Authentication** > **Users**.
   b. For **Add**, type a name for the user (for example, **OpenVPN_User**) and click ✚.



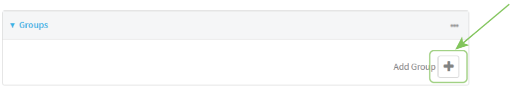
Add OpenVPN_User

   c. Type a password for the user.

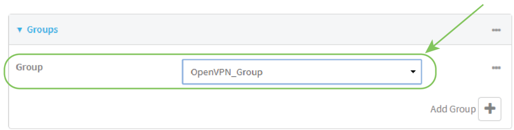
      This password is used for local authentication of the user. You can also configure the user to use RADIUS or TACACS+ authentication by configuring authentication methods. See User authentication methods for information.
   d. Click to expand the **Groups** node.



   e. Click ✚ to add a group to the user.

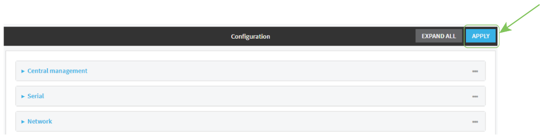


   f. Select a **Group** with **OpenVPN access** enabled.



Group OpenVPN_Group

5. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. Use the **add auth group** command to add a new authentication. For example, to add a group named **OpenVPN_Group**:

   ```
   (config)> add auth group OpenVPN_Group
   (config auth group OpenVPN_Group)>
   ```

4. Enable OpenVPN access rights for users of this group:

   ```
   (config auth group OpenVPN_Group)> acl openvpn enable true
   ```

5. Add an OpenVPN tunnel to which users of this group will have access:

   a. Determine available tunnels:

   ```
   (config auth group OpenVPN_Group)> .. .. .. vpn openvpn server ?

   Servers: A list of openvpn servers

    Additional Configuration
    -----------------------------------------------------------------------
   --------
    OpenVPN_server1           OpenVPN server

   (config auth group OpenVPN_Group)>
   ```

   b. Add a tunnel:

   ```
   (config auth group OpenVPN_Group)> add auth group test acl openvpn
   tunnels end /vpn/openvpn/server/OpenVPN_server1
   (config auth group OpenVPN_Group)>
   ```

6. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure an OpenVPN client by using an .ovpn file

Required configuration items

- Enable the OpenVPN client.

  The OpenVPN client is enabled by default.
- The firewall zone to be used by the OpenVPN client.

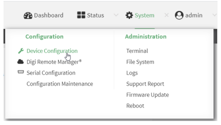
Additional configuration items

- The route metric for the OpenVPN client.
- The login credentials for the OpenVPN client, if configured on the OpenVPN server.

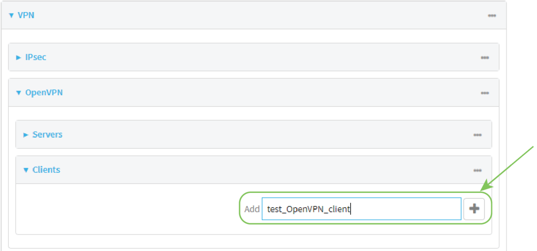
See Configure SureLink active recovery for OpenVPN for information about OpenVPN active recovery.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

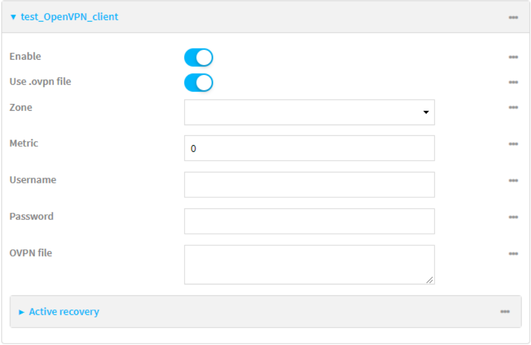


   The **Configuration** window is displayed.
3. Click **VPN** > **OpenVPN** > **Clients**.
4. For **Add**, type a name for the OpenVPN client and click ✚.



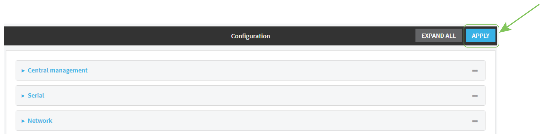
Add test_OpenVPN_client

   The new OpenVPN client configuration is displayed.



Metric 0

5. The OpenVPN client is enabled by default. To disable, click **Enable**.
6. The default behavior is to use an OVPN file for client configuration. To disable this behavior and configure the client manually, click **Use .ovpn file** to disable. If **Use .ovpn file** is disabled, see Configure an OpenVPN client without using an .ovpn file for configuration information.
7. For **Zone**, select the firewall zone for the OpenVPN client.
8. (Optional) Select the **Metric** for the OpenVPN client. If multiple active routes match a destination, the route with the lowest metric will be used.
9. (Optional) For **Username** and **Password**, type the login credentials as configured on the OpenVPN server.
10. For **OVPN file**, paste the content of the client.ovpn file.
11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. At the config prompt, type:

```
(config)> add vpn openvpn client name
(config vpn openvpn client name)>
```

   where *name* is the name of the OpenVPN server.

   The OpenVPN client is enabled by default. To disable the client, type:

```
(config vpn openvpn client name)> enable false
(config vpn openvpn client name)>
```

4. Set the firewall zone for the OpenVPN client:

```
(config vpn openvpn client name)> zone value
(config vpn openvpn client name)>
```

   To view a list of available zones:

```
(config vpn openvpn client name)> zone ?

Zone: The zone for the openvpn client interface.
Format:
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup
Current value:

(config vpn openvpn client name)>
```

5. (Optional) Set the route metric for the OpenVPN server. If multiple active routes match a destination, the route with the lowest metric will be used.

```
(config vpn openvpn client name)> metric value
(config vpn openvpn client name)>
```

   where *value* is an interger between **0** and **65535**. The default is **0**.

6. (Optional) Set the login credentials as configured on the OpenVPN server:

```
(config vpn openvpn client name)> username value
(config vpn openvpn client name)> password value
(config vpn openvpn client name)>
```

7. Paste the content of the client.ovpn file into the value of the **config_file** parameter:

```
(config vpn openvpn client name)> config_file value
(config vpn openvpn client name)>
```

8. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure an OpenVPN client without using an .ovpn file

Required configuration items

- Enable the OpenVPN client.

  The OpenVPN client is enabled by default.
- The mode used by the OpenVPN server, either routing (TUN), or bridging (TAP).
- The firewall zone to be used by the OpenVPN client.
- The IP address of the OpenVPN server.

- Certificates and keys:
  - The **CA certificate** (usually in a ca.crt file).
  - The **Public key** (for example, client.crt)
  - The **Private key** (for example, client.key).

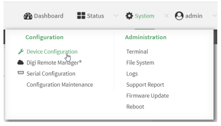
Additional configuration items

- The route metric for the OpenVPN client.
- The login credentials for the OpenVPN client, if configured on the OpenVPN server.
- Additional OpenVPN parameters.

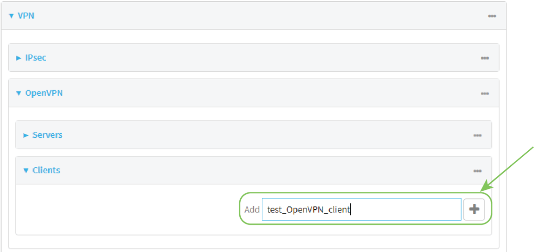
See Configure SureLink active recovery for OpenVPN for information about OpenVPN active recovery.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

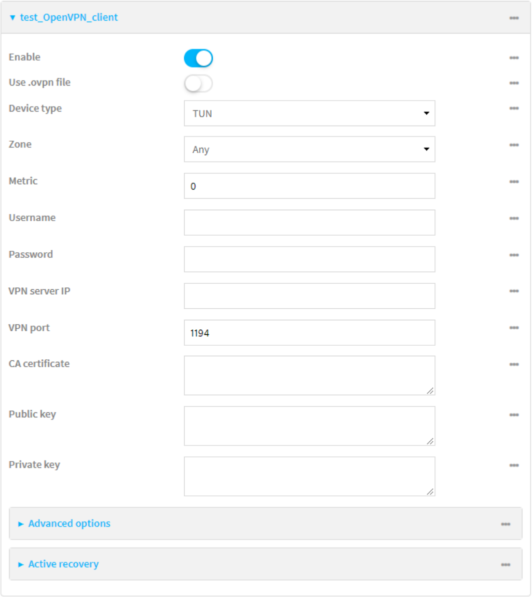


   The **Configuration** window is displayed.
3. Click **VPN** > **OpenVPN** > **Clients**.
4. For **Add**, type a name for the OpenVPN client and click ✚.



Add test_OpenVPN_client

   The new OpenVPN client configuration is displayed.



▾ test_OpenVPN_client

Enable: ●

Use .ovpn file: ○

Device type: TUN

Zone: Any

Metric: 0

Username:

Password:

VPN server IP:

VPN port: 1194

CA certificate:

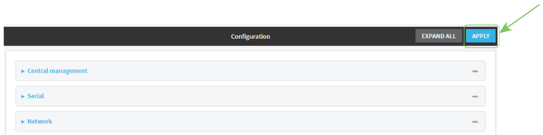
Public key:

Private key:

▸ Advanced options

▸ Active recovery

5. The OpenVPN client is enabled by default. To disable, click **Enable**.
6. The default behavior is to use an OVPN file for client configuration. To disable this behavior and configure the client manually, click **Use .ovpn file** to disable.
7. For **Device type**, select the mode used by the OpenVPN server, either **TUN** or **TAP**.
8. For **Zone**, select the firewall zone for the OpenVPN client.
9. (Optional) Select the **Metric** for the OpenVPN client. If multiple active routes match a destination, the route with the lowest metric will be used.
10. (Optional) For **Username** and **Password**, type the login credentials as configured on the OpenVPN server.
11. For **VPN server IP**, type the IP address of the OpenVPN server.
12. (Optional) Set the **VPN port** used by the OpenVPN server. The default is **1194**.
13. Paste the contents of the **CA certificate** (usually in a ca.crt file), the **Public key** (for example, client.crt), and the **Private key** (for example, client.key) into their respective fields. The contents will be hidden when the configuration is saved.
14. (Optional) Click to expand **Advanced Options** to manually set additional OpenVPN parameters.
    a. Click **Enable** to enable the use of additional OpenVPN parameters.
    b. Click **Override** if the additional OpenVPN parameters should override default options.
    c. For **OpenVPN parameters**, type the additional OpenVPN parameters. For example, to override the configuration by using a configuration file, enter **--config filename**, for example, **--config /etc/config/openvpn_config**.
15. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. At the config prompt, type:

```
(config)> add vpn openvpn client name
(config vpn openvpn client name)>
```

   where *name* is the name of the OpenVPN server.

   The OpenVPN client is enabled by default. To disable the client, type:

```
(config vpn openvpn client name)> enable false
(config vpn openvpn client name)>
```

4. The default behavior is to use an OVPN file for client configuration. To disable this behavior and configure the client manually:

```
(config vpn openvpn client name)> use_file false
(config vpn openvpn client name)>
```

5. Set the mode used by the OpenVPN server:

```
(config vpn openvpn client name)> device_type value
(config vpn openvpn client name)>
```

   where *value* is either **tun** or **tap**. The default is **tun**.
6. Set the firewall zone for the OpenVPN client:

```
(config vpn openvpn client name)> zone value
(config vpn openvpn client name)>
```

   To view a list of available zones:

```
(config vpn openvpn client name)> zone ?

Zone: The zone for the openvpn client interface.
Format:
  any
  dynamic_routes
  edge
  external
  internal
  ipsec
  loopback
  setup
```

```
Current value:

(config vpn openvpn client name)>
```

7. (Optional) Set the route metric for the OpenVPN server. If multiple active routes match a destination, the route with the lowest metric will be used.

   ```
   (config vpn openvpn client name)> metric value
   (config vpn openvpn client name)>
   ```

   where *value* is an interger between **0** and **65535**. The default is **0**.

8. (Optional) Set the login credentials as configured on the OpenVPN server:

   ```
   (config vpn openvpn client name)> username value
   (config vpn openvpn client name)> password value
   (config vpn openvpn client name)>
   ```

9. Set the IP address of the OpenVPN server:

   ```
   (config vpn openvpn client name)> server ip_address
   (config vpn openvpn client name)>
   ```

10. (Optional) Set the port used by the OpenVPN server:

    ```
    (config vpn openvpn client name)> port port
    (config vpn openvpn client name)>
    ```

    The default is **1194**.

11. Paste the contents of the CA certificate (usually in a ca.crt file) into the value of the **cacert** parameter:

    ```
    (config vpn openvpn client name)> cacert value
    (config vpn openvpn client name)>
    ```

12. Paste the contents of the public key (for example, client.crt) into the value of the **public_cert** parameter:

    ```
    (config vpn openvpn client name)> public_cert value
    (config vpn openvpn client name)>
    ```

13. Paste the contents of the private key (for example, client.key) into the value of the **private_key** parameter:

    ```
    (config vpn openvpn client name)> private_key value
    (config vpn openvpn client name)>
    ```

14. (Optional) Set additional OpenVPN parameters.

    a. Enable the use of additional OpenVPN parameters:

       ```
       (config vpn openvpn client name)> advanced_options enable true
       (config vpn openvpn client name)>
       ```

b. Configure whether the additional OpenVPN parameters should override default options:

```
(config vpn openvpn client name)> advanced_options override true
(config vpn openvpn client name)>
```

c. Set the additional OpenVPN parameters:

```
(config vpn openvpn client name)> advanced_options extra parameters
(config vpn openvpn client name)>
```

15. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

16. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure SureLink active recovery for OpenVPN

You can configure the AnywhereUSB Plus device to regularly probe OpenVPN client connections to determine if the connection has failed and take remedial action.

Required configuration items

- A valid OpenVPN client configuration. See Configure an OpenVPN client by using an .ovpn file or Configure an OpenVPN client without using an .ovpn file for configuration instructions.
- Enable OpenVPN active recovery.
- The behavior of the AnywhereUSB Plus device upon OpenVPN failure: either
  - Restart the OpenVPN interface
  - Reboot the device.

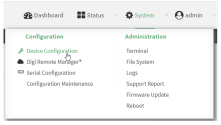
Additional configuration items

- The interval between connectivity tests.
- Whether the interface should be considered to have failed if one of the test targets fails, or all of the test targets fail.
- The number of probe attempts before the OpenVPN connection is considered to have failed.
- The amount of time that the device should wait for a response to a probe attempt before considering it to have failed.

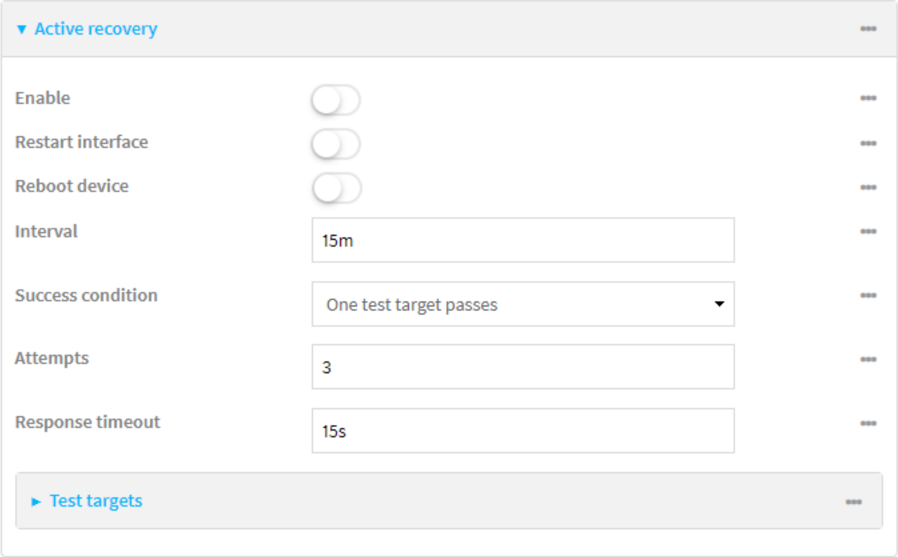
To configure the AnywhereUSB Plus device to regularly probe the OpenVPN connection:

☰ **WebUI**

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **VPN** > **OpenVPN** > **Clients**.
4. Create a new OpenVPN client or select an existing one:
   - To create a new OpenVPN client, see Configure an OpenVPN client by using an .ovpn file or Configure an OpenVPN client without using an .ovpn file.
   - To edit an existing OpenVPN client, click to expand the appropriate client.
5. After creating or selecting the OpenVPN client, click **Active recovery**.

   ▼ Active recovery

   Enable

   Restart interface

   Reboot device

   Interval 15m

   Success condition One test target passes

   Attempts 3

   Response timeout 15s

   ► Test targets

6. **Enable** active recovery.
7. For **Restart interface**, enable to configure the device to restart the interface when its connection is considered to have failed. This is useful for interfaces that may regain connectivity after restarting, such as a cellular modem.
8. For **Reboot device**, enable to instruct the device to reboot when the WAN connection is considered to have failed.
9. Change the **Interval** between connectivity tests.

   Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **Interval** to ten minutes, enter **10m** or **600s**.

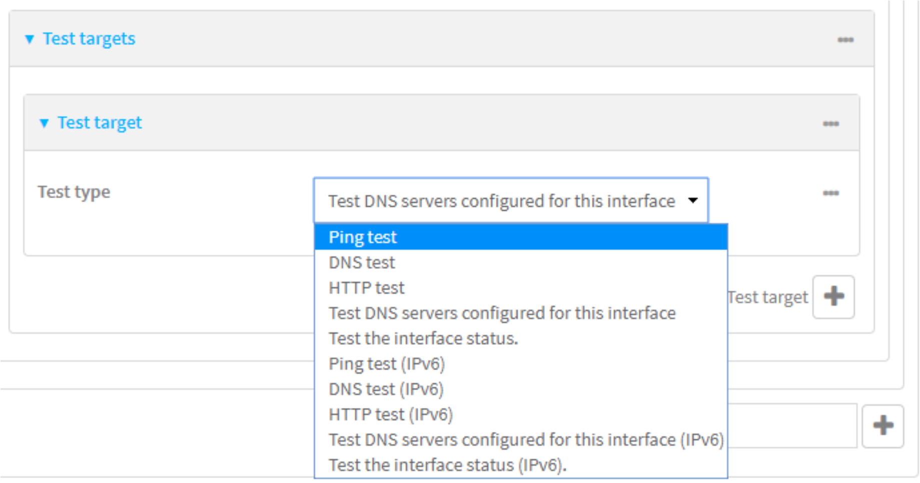
   The default is 15 minutes.
10. For **Success condition**, determine whether the interface should fail over based on the failure of one of the test targets, or all of the test targets.

11. For **Attempts**, type the number of probe attempts before the WAN is considered to have failed.
12. For **Response timeout**, type the amount of time that the device should wait for a response to a probe attempt before considering it to have failed.

    Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **Response timeout** to ten minutes, enter **10m** or **600s**.

    The default is 15 seconds.
13. Add a test target:
    a. Click to expand **Test targets**.


▾ Test targets
▾ Test target
Test type
Test DNS servers configured for this interface
Ping test
DNS test
HTTP test
Test DNS servers configured for this interface
Test the interface status.
Ping test (IPv6)
DNS test (IPv6)
HTTP test (IPv6)
Test DNS servers configured for this interface (IPv6)
Test the interface status (IPv6).
Test target

    b. For **Add Test target**, click **+**.
    c. Select the **Test type**:

- **Ping test** or **Ping test (IPv6)**: Tests connectivity by sending an ICMP echo request to the hostname or IP address specified in **Ping host**. You can also optionally change the number of bytes in the **Ping payload size**.
- **DNS test** or **DNS test (IPv6)**: Tests connectivity by sending a DNS query to the specified **DNS server**.
- **HTTP test HTTP test (IPv6)**: Tests connectivity by sending an HTTP or HTTPS GET request to the URL specified in **Web servers**. The URL should take the format of **http[s]://*hostname*/[*path*]**.
- **Test DNS servers configured for this interface** or **Test DNS servers configured for this interface (IPv6)**: Tests connectivity by sending a DNS query to the DNS servers configured for this interface.
- **Test the interface status** or **Test the interface status IPv6**: The interface is considered to be down based on:
    - **Down time**: The amount of time that the interface can be down before this test is considered to have failed.

      Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

      For example, to set **Down time** to ten minutes, enter **10m** or **600s**.

  The default is 60 seconds.

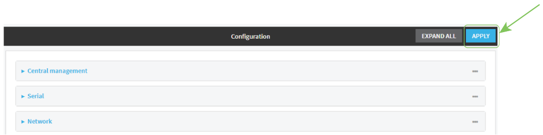
- **Initial connection time**: The amount of time to wait for an initial connection to the interface before this test is considered to have failed.

  Allowed values are any number of weeks, days, hours, minutes, or seconds, and take the format ***number***{**w**|**d**|**h**|**m**|**s**}.

  For example, to set **Initial connection time** to ten minutes, enter **10m** or **600s**.

  The default is 60 seconds.

14. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Create a new OpenVPN client, or edit an existing one:
   - To create a new OpenVPN client, see Configure an OpenVPN client by using an .ovpn file or Configure an OpenVPN client without using an .ovpn file.
   - To edit an existing OpenVPN client, change to the OpenVPN client's node in the configuration schema. For example, for an OpenVPN client named **openvpn_client1**, change to the **openvpn_client1** node in the configuration schema:

```
(config)> vpn openvpn client openvpn_client1
(config vpn openvpn client openvpn_client1)>
```

4. Enable active recovery:

```
(config vpn openvpn client openvpn_client1)> connection_monitor enable true
(config vpn openvpn client openvpn_client1)>
```

5. To configure the device to restart the interface when its connection is considered to have failed:

```
(config vpn openvpn client openvpn_client1)> connection_monitor restart
true
(config vpn openvpn client openvpn_client1)>
```

   This is useful for interfaces that may regain connectivity after restarting, such as a cellular modem.

6. To configure the device to reboot when the interface is considered to have failed:

   ```
   (config vpn openvpn client openvpn_client1)> connection_monitor reboot
   enable
   (config vpn openvpn client openvpn_client1)>
   ```

7. Set the **Interval** between connectivity tests:

   ```
   (config vpn openvpn client openvpn_client1)> connection_monitor interval
   value
   (config vpn openvpn client openvpn_client1)>
   ```

   where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

   For example, to set **interval** to ten minutes, enter either **10m** or **600s**:

   ```
   (config vpn openvpn client openvpn_client1)> connection_monitor interval
   600s
   (config vpn openvpn client openvpn_client1)>
   ```

   The default is 15 minutes.

8. Determine whether the interface should fail over based on the failure of one of the test targets, or all of the test targets:

   ```
   (config vpn openvpn client openvpn_client1)> connection_monitor success_
   condition value
   (config vpn openvpn client openvpn_client1)>
   ```

   Where *value* is either **one** or **all**.

9. Set the number of probe attempts before the WAN is considered to have failed:

   ```
   (config vpn openvpn client openvpn_client1)> connection_monitor attempts
   num
   (config vpn openvpn client openvpn_client1)>
   ```

   The default is **3**.

10. Set the amount of time that the device should wait for a response to a probe attempt before considering it to have failed:

    ```
    (config vpn openvpn client openvpn_client1)> connection_monitor timeout
    value
    (config vpn openvpn client openvpn_client1)>
    ```

    where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w**|**d**|**h**|**m**|**s**}.

    For example, to set **timeout** to ten minutes, enter either **10m** or **600s**:

    ```
    (config vpn openvpn client openvpn_client1)> connection_monitor interval
    600s
    (config vpn openvpn client openvpn_client1)>
    ```

    The default is 15 seconds.

11. Configure test targets:

    a. Add a test target:

    ```
    (config vpn openvpn client openvpn_client1)> add connection_monitor
    target end
    (config vpn openvpn client openvpn_client1 connection_monitor target 0)>
    ```

    b. Set the test type:

    ```
    (config vpn openvpn client openvpn_client1 connection_monitor target 0)>
    test value
    (config vpn openvpn client openvpn_client1 connection_monitor target 0)>
    ```

    where *value* is one of:

    - **ping** (IPv4) or **ping6** (IPv6): Tests connectivity by sending an ICMP echo request to a specified hostname or IP address.
        - Specify the hostname or IP address by using **ping_host** or **ping_host6**:

        ```
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)> ping_host host
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)>
        ```

        - (Optional) Set the size, in bytes, of the ping packet by using **ping_size** or **ping_size6**:

        ```
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)> ping_size [num]
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)>
        ```

    - **dns** (IPv4) or **dns6** (IPv6): Tests connectivity by sending a DNS query to the specified DNS server.
        - Specify the DNS server. Allowed value is the IP address of the DNS server.

        ```
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)> dns_server ip_address
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)>
        ```

    - **dns_configured** (IPv4) or **dns_configured6** (IPv6): Tests connectivity by sending a DNS query to the DNS servers configured for this interface.
    - **http** (IPv4) or **http6** (IPv6): Tests connectivity by sending an HTTP or HTTPS GET request to the specified URL.
        - Specify the url. Allowed value uses the format **http[s]://*hostname*/[*path*]**.

        ```
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)> http_url url
        (config vpn openvpn client openvpn_client1 connection_monitor
        target 0)>
        ```

- **interface_up** (IPv4) or **interface_up6** (IPv6): : The interface is considered to be down based on the interfaces down time, and the amount of time an initial connection to the interface takes before this test is considered to have failed.
    - (Optional) Set the amount of time that the interface can be down before this test is considered to have failed:

      ```
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)> interface_down_time value
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)>
      ```

      where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

      For example, to set **interface_down_time** to ten minutes, enter either **10m** or **600s**:

      ```
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)> interface_down_time 600s
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)>
      ```

      The default is 60 seconds.
    - (Optional) Set the amount of time to wait for an initial connection to the interface before this test is considered to have failed:

      ```
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)> interface_timeout value
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)>
      ```

      where *value* is any number of weeks, days, hours, minutes, or seconds, and takes the format ***number***{**w|d|h|m|s**}.

      For example, to set **interface_timeout** to ten minutes, enter either **10m** or **600s**:

      ```
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)> interface_timeout 600s
      (config vpn openvpn client openvpn_client1 connection_monitor
      target 0)>
      ```

      The default is 60 seconds.

12. Save the configuration and apply the change:

    ```
    (config vpn openvpn client openvpn_client1 connection_monitor target 0)>
    save
    Configuration saved.
    >
    ```

13. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show OpenVPN server status and statistics

You can view status and statistics for OpenVPN servers from either the web interface or the command line:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, select **Status** > **OpenVPN** > **Servers**.

   The **OpenVPN Servers** page appears.
3. To view configuration details about an OpenVPN server, click the 🔧 (configuration) icon in the upper right of the OpenVPN server's status pane.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. To display details about all configured OpenVPN servers, type the following at the prompt:

```
> show openvpn server all

 Server           Enable  Type  Zone      Address          Port
 ---------------  ------  ----  --------  ---------------  ----
 OpenVPN_server1  true    tun   internal  192.168.30.1/24  1194
 OpenVPN_server2  false   tun   internal  192.168.40.1/24  1194

>
```

3. To display details about a specific server:

```
> show openvpn server name OpenVPN_server1

 Server                    : OpenVPN_server1
 Enable                    : true
 Type                      : tun
 Zone                      : internal
 Address                   : 192.168.30.1/24
 Port                      : 1194
 Use File                  : true
 Metric                    : 0
 Protocol                  : udp
 First IP                  : 80
 Last IP                   : 99

>
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show OpenVPN client status and statistics

You can view status and statistics for OpenVPN clients from either web interface or the command line:

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, select **Status** > **OpenVPN** > **Clients**.

   The **OpenVPN Clients** page appears.
3. To view configuration details about an OpenVPN client, click the 🔧 (configuration) icon in the upper right of the OpenVPN client's status pane.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. To display details about all configured OpenVPN clients, type the following at the prompt:

```
> show openvpn client all

 Client           Enable  Status     Username  Use File  Zone
 ---------------  ------  -------    --------  --------  --------
 OpenVPN_Client1  true    connected            true      internal
 OpenVPN_Client2  true    pending            true      internal

>
```

3. To display details about a specific server:

```
> show openvpn client name OpenVPN_client1
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Generic Routing Encapsulation (GRE)

Generic Routing Encapsulation (GRE) is an IP packet encapsulation protocol that allow for networks and routes to be advertized from one network device to another. You can use GRE to encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an IP network.

Configuring a GRE tunnel

Configuring a GRE tunnel involves the following items:

Required configuration items

- A GRE loopback endpoint interface.
- GRE tunnel configuration:
  - Enable the GRE tunnel.

    The GRE tunnels are enabled by default.
  - The local endpoint interface.
  - The IP address of the remote device/peer.

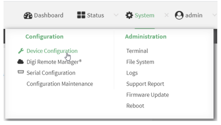
Additional configuration items

- A GRE key.
- Enable the device to respond to keepalive packets.

Task One: Create a GRE loopback endpoint interface

WebUI

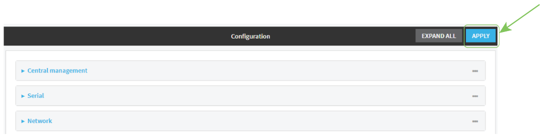
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **Network** > **Interfaces**.
4. For **Add Interface**, type a name for the GRE loopback endpoint interface and click ✚.
5. **Enable** the interface.

   New interfaces are enabled by default. To disable, or to enable if it has been disabled, click **Enable**.
6. For **Interface type**, select **Ethernet**.
7. For **Zone**, select **Internal**.
8. For **Device**, select **Ethernet: Loopback**.

9. Click to expand **IPv4**.
10. For **Address**, enter the IP address and subnet mask of the local GRE endpoint, for example **10.10.1.1/24**.
11. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add the GRE endpoint interface. For example, to add an interface named **gre_endpoint**:

```
(config)> add network interface gre_interface
(config network interface gre_interface)>
```

4. Set the interface zone to **internal**:

```
(config network interface gre_interface)> zone internal
(config network interface gre_interface)>
```

5. Set the interface device to **loopback**:

```
(config network interface gre_interface)> device /network/device/loopback
(config network interface gre_interface)>
```

6. Set the IP address and subnet mask of the local GRE endpoint. For example, to set the local GRE endpoint's IP address and subnet mask to **10.10.1.1/24:**

```
(config network interface gre_interface)> ipv4 address 10.10.1.1/24
(config network interface gre_interface)>
```

7. Save the configuration and apply the change:

```
(config network interface gre_interface)> save
Configuration saved.
>
```

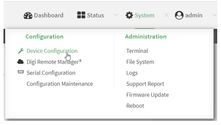
8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Task Two: Configure the GRE tunnel

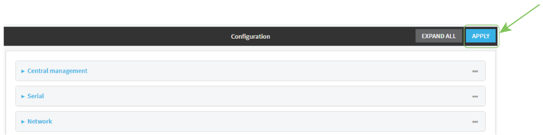
WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **VPN** > **IP Tunnels**.
4. For **Add IP tunnel**, type a name for the GRE tunnel and click **+**.
5. **Enable** the tunnel.

   New tunnels are enabled by default. To disable, or to enable if it has been disabled, click **Enable**.
6. For **Local endpoint**, select the GRE endpoint interface created in Task One.
7. For **Remote endpoint**, type the IP address of the GRE endpoint on the remote peer.
8. (Optional) For **Key**, enter a key that will be inserted in GRE packets created by this tunnel. It must match the key set by the remote endpoint. Allowed value is an interger between 0 and 4294967295, or an IP address.
9. (Optional) **Enable keepalive reply** to enable the device to reply to Cisco GRE keepalive packets.
10. Click **Apply** to save the configuration and apply the change.

Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add the GRE endpoint tunnel. For example, to add a tunnel named **gre_example**:

   ```
   (config)> add vpn iptunnel gre_example
   (config vpn iptunnel gre_example)>
   ```

GRE tunnels are enabled by default. To disable:

```
(config vpn iptunnel gre_example)> enable false
(config vpn iptunnel gre_example)>
```

4. Set the local endpoint to the GRE endpoint interface created in Task One, for example:

```
(config vpn iptunnel gre_example)> local /network/interface/gre_endpoint
(config vpn iptunnel gre_example)>
```

5. Set the IP address of the GRE endpoint on the remote peer:

```
(config vpn iptunnel gre_example)> remote ip_address
(config vpn iptunnel gre_example)>
```

6. (Optional) Set a key that will be inserted in GRE packets created by this tunnel.

   The key must match the key set by the remote endpoint.

```
(config vpn iptunnel gre_example)> key value
(config vpn iptunnel gre_example)>
```

   where value is an interger between 0 and 4294967295, or an IP address.

7. (Optional) Enable the device to reply to Cisco GRE keepalive packets:

```
(config vpn iptunnel gre_example)> keepalive true
(config vpn iptunnel gre_example)>
```

8. Save the configuration and apply the change:

```
(config vpn iptunnel gre_example)> save
Configuration saved.
>
```

9. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show GRE tunnels

To view information about currently configured GRE tunnels:

☰ WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, click **Status** > **IP tunnels**.

   The **IP Tunnels**page appears.
3. To view configuration details about a GRE tunnel, click the 🔧 (configuration) icon in the upper right of the tunnel's status pane.

Example: GRE tunnel over an IPSec tunnel

The AnywhereUSB Plus device can be configured as an advertised set of routes through an IPSec tunnel. This allows you to leverage the dynamic route advertisement of GRE tunnels through a secured IPSec tunnel.

The example configuration provides instructions for configuring the AnywhereUSB Plus device with a GRE tunnel through IPsec.

AnywhereUSB Plus-1 configuration tasks

1. Create an IPsec tunnel named **ipsec_gre1** with:
   - A pre-shared key.
   - **Remote endpoint** set to the public IP address of the AnywhereUSB Plus-2 device.
   - A policy with:
     - **Local network** set to the IP address and subnet of the local GRE tunnel, **172.30.0.1/32**.
     - **Remote network** set to the IP address and subnet of the remote GRE tunnel, **172.30.0.2/32**.
2. Create an IPsec endpoint interface named **ipsec_endpoint1**:
   a. **Zone** set to **Internal**.
   b. **Device** set to **Ethernet: Loopback**.
   c. IPv4 Address set to the IP address of the local GRE tunnel, **172.30.0.1/32**.
3. Create a GRE tunnel named **gre_tunnel1**:
   a. **Local endpoint** set to the IPsec endpoint interface, **Interface: ipsec_endpoint1**.
   b. Remote endpoint set to the IP address of the GRE tunnel on AnywhereUSB Plus-2, **172.30.0.2**.
4. Create an interface named **gre_interface1** and add it to the GRE tunnel:
   a. **Zone** set to **Internal**.
   b. **Device** set to **IP tunnel: gre_tunnel1**.
   c. IPv4 Address set to a virtual IP address on the GRE tunnel, **172.31.0.1/30**.

AnywhereUSB Plus-2 configuration tasks

1. Create an IPsec tunnel named **ipsec_gre2** with:
   - The same pre-shared key as the **ipsec_gre1** tunnel on AnywhereUSB Plus-1.
   - **Remote endpoint** set to the public IP address of AnywhereUSB Plus-1.
   - A policy with:
     - **Local network** set to the IP address and subnet of the local GRE tunnel, **172.30.0.2/32**.
     - **Remote network** set to the IP address of the remote GRE tunnel, **172.30.0.1/32**.
2. Create an IPsec endpoint interface named **ipsec_endpoint2**:
   a. **Zone** set to **Internal**.
   b. **Device** set to **Ethernet: Loopback**.
   c. IPv4 Address set to the IP address of the local GRE tunnel, **172.30.0.2/32**.

3. Create a GRE tunnel named **gre_tunnel2**:
   a. **Local endpoint** set to the IPsec endpoint interface, **Interface: ipsec_endpoint2**.
   b. Remote endpoint set to the IP address of the GRE tunnel on AnywhereUSB Plus-1, **172.30.0.1**.
4. Create an interface named **gre_interface2** and add it to the GRE tunnel:
   a. **Zone** set to **Internal**.
   b. **Device** set to **IP tunnel: gre_tunnel2**.
   c. IPv4 Address set to a virtual IP address on the GRE tunnel, **172.31.1.1/30**.

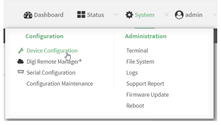
Configuration procedures

Configure the AnywhereUSB Plus-1 device

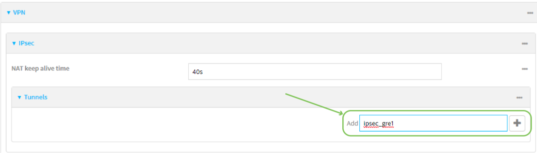
Task one: Create an IPsec tunnel

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



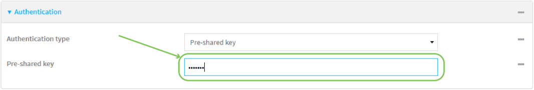
   The **Configuration** window is displayed.
3. Click **VPN** > **IPsec** > **Tunnels**.
4. For **Add IPsec Tunnel**, type **ipsec_gre1** and click **+**.



NAT keep alive time 40s

Add ipsec_gre1

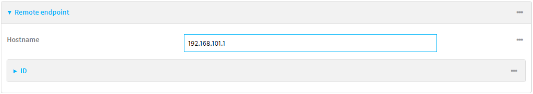
5. Click to expand **Authentication**.
6. For **Pre-shared key**, type **testkey**.



Authentication type Pre-shared key

Pre-shared key •••••••

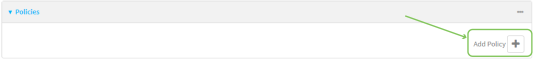
7. Click to expand **Remote endpoint**.
8. For **Hostname**, type public IP address of the AnywhereUSB Plus-2 device.



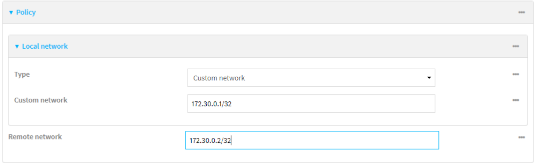
Hostname 192.168.101.1

9. Click to expand **Policies**.

10. For **Add Policy**, click ✚ to add a new policy.



11. Click to expand **Local network**.
12. For **Type**, select **Custom network**.
13. For **Address**, type the IP address and subnet of the local GRE tunnel, **172.30.0.1/32**.
14. For **Remote network**, type the IP address and subnet of the remote GRE tunnel, **172.30.0.2/32**.

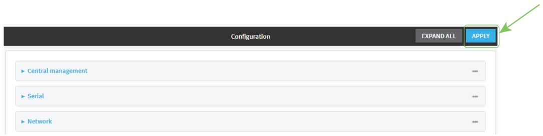


Type: Custom network

Custom network: 172.30.0.1/32

Remote network: 172.30.0.2/32

15. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add an IPsec tunnel named **ipsec_gre1**:

```
(config)> add vpn ipsec tunnel ipsec_gre1
(config vpn ipsec tunnel ipsec_gre1)>
```

4. Set the pre-shared key to **testkey**:

```
(config vpn ipsec tunnel ipsec_gre1)> auth secret testkey
(config vpn ipsec tunnel ipsec_gre1)>
```

5. Set the remote endpoint to public IP address of the AnywhereUSB Plus-2 device:

```
(config vpn ipsec tunnel ipsec_gre1)> remote hostname 192.168.101.1
(config vpn ipsec tunnel ipsec_gre1)>
```

6. Add a policy:

   ```
   (config vpn ipsec tunnel ipsec_gre1)> add policy end
   (config vpn ipsec tunnel ipsec_gre1 policy 0)>
   ```

7. Set the local network policy type to **custom**:

   ```
   (config vpn ipsec tunnel ipsec_gre1 policy 0)> local type custom
   (config vpn ipsec tunnel ipsec_gre1 policy 0)>
   ```

8. Set the local network address to the IP address and subnet of the local GRE tunnel, **172.30.0.1/32**:

   ```
   (config vpn ipsec tunnel ipsec_gre1 policy 0)> local custom 172.30.0.1/32
   (config vpn ipsec tunnel ipsec_gre1 policy 0)>
   ```

9. Set the remote network address to the IP address and subnet of the remote GRE tunnel, **172.30.0.2/32**:

   ```
   (config vpn ipsec tunnel ipsec_gre1 policy 0)> remote network 172.30.0.2/32
   (config vpn ipsec tunnel ipsec_gre1 policy 0)>
   ```

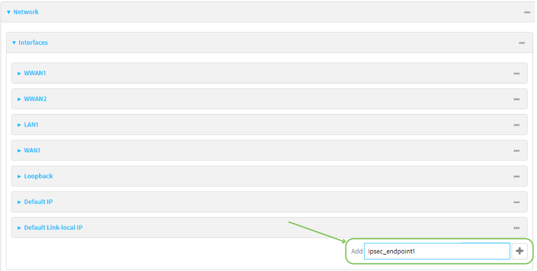
10. Save the configuration and apply the change:

    ```
    (config ipsec tunnel ipsec_gre1 policy 0)> save
    Configuration saved.
    >
    ```

Task two: Create an IPsec endpoint interface

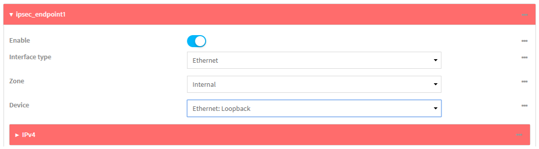
WebUI

1. Click **Network** > **Interface**.
2. For **Add Interface**, type **ipsec_endpoint1** and click **+**.



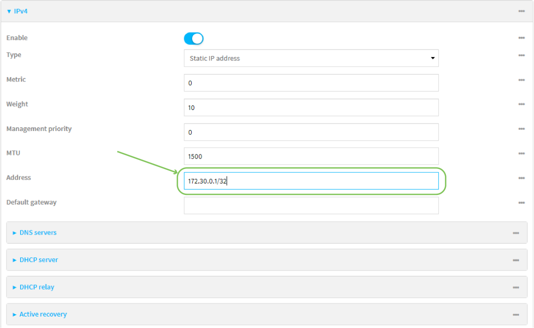
Add ipsec_endpoint1

3. For **Zone**, select **Internal**.
4. For **Device**, select **Ethernet: loopback**.



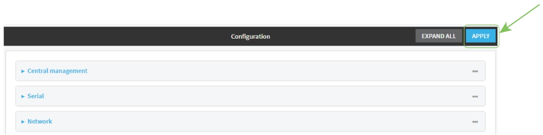
Enable ●
Interface type Ethernet
Zone Internal
Device Ethernet: Loopback

5. Click to expand **IPv4**.
6. For **Address**, type the IP address of the local GRE tunnel, **172.30.0.1/32**.



Enable ●
Type Static IP address
Metric 0
Weight 10
Management priority 0
MTU 1500
Address 172.30.0.1/32
Default gateway

7. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add an interface named **ipsec_endpoint1**:

```
(config)> add network interface ipsec_endpoint1
(config network interface ipsec_endpoint1)>
```

3. Set the zone to **internal**:

```
(config network interface ipsec_endpoint1)> zone internal
(config network interface ipsec_endpoint1)>
```

4. Set the device to **/network/device/loopback**:

```
(config network interface ipsec_endpoint1)> device /network/device/loopback
(config network interface ipsec_endpoint1)>
```

5. Set the IPv4 address to the IP address of the local GRE tunnel, **172.30.0.1/32**:

```
(config network interface ipsec_endpoint1)> ipv4 address 172.30.0.1/32
(config network interface ipsec_endpoint1)>
```

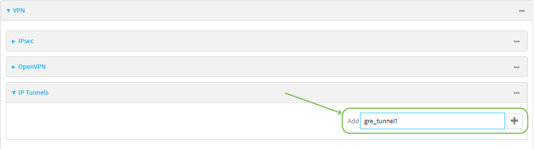
6. Save the configuration and apply the change:

```
(config vpn ipsec tunnel ipsec_endpoint1 policy 0)> save
Configuration saved.
>
```

Task three: Create a GRE tunnel

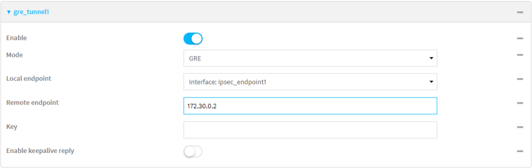
WebUI

1. Click **VPN** > **IP Tunnels**.
2. For **Add IP Tunnel**, type **gre_tunnel1** and click **+**.



gre_tunnel1

3. For **Local endpoint**, select the IPsec endpoint interface created in Task two (**Interface: ipsec_endpoint1**).
4. For **Remote endpoint**, type the IP address of the GRE tunnel on AnywhereUSB Plus-2, **172.30.0.2**.



▾ gre_tunnel1

Enable: on

Mode: GRE

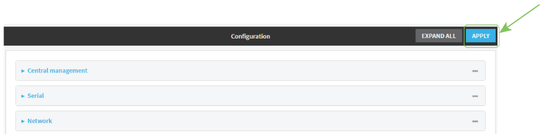
Local endpoint: Interface: ipsec_endpoint1

Remote endpoint: 172.30.0.2

Key:

Enable keepalive reply: off

5. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add a GRE tunnel named **gre_tunnel1**:

```
(config)> add vpn iptunnel gre_tunnel1
(config vpn iptunnel gre_tunnel1)>
```

3. Set the local endpoint to the IPsec endpoint interface created in Task two (**/network/interface/ipsec_endpoint1**):

```
(config vpn iptunnel gre_tunnel1)> local /network/interface/ipsec_endpoint1
(config vpn iptunnel gre_tunnel1)>
```

4. Set the remote endpoint to the IP address of the GRE tunnel on AnywhereUSB Plus-2, **172.30.0.2**:

```
(config vpn iptunnel gre_tunnel1)> remote 172.30.0.2
(config vpn iptunnel gre_tunnel1)>
```

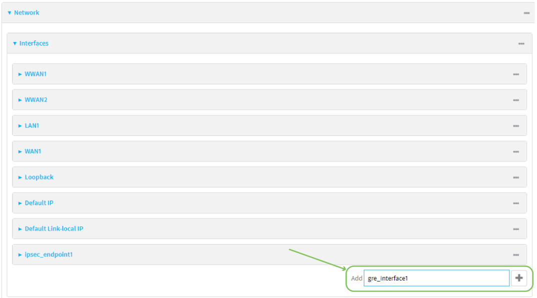
5. Save the configuration and apply the change:

```
(config vpn iptunnel gre_tunnel1)> save
Configuration saved.
>
```

Task four: Create an interface for the GRE tunnel device

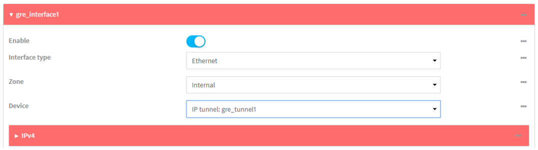
WebUI

1. Click **Network** > **Interfaces**.
2. For **Add Interface**, type **gre_interface1** and click **+**.



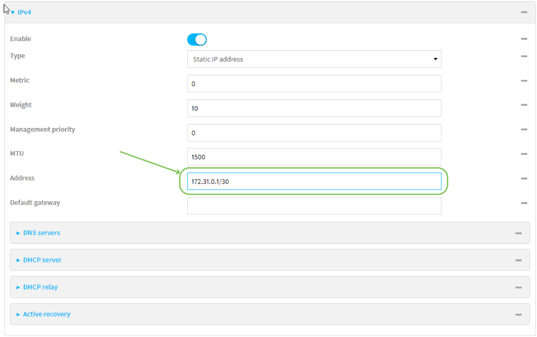
Add gre_interface1

3. For **Zone**, select **Internal**.
4. For **Device**, select the GRE tunnel created in Task three (**IP tunnel: gre_tunnel1**).



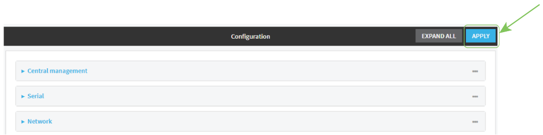
Enable
Interface type: Ethernet
Zone: Internal
Device: IP tunnel: gre_tunnel1

5. Click to expand **IPv4**.
6. For **Address**, type **172.31.0.1/30** for a virtual IP address on the GRE tunnel.



Enable
Type: Static IP address
Metric: 0
Weight: 10
Management priority: 0
MTU: 1500
Address: 172.31.0.1/30
Default gateway

7. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add an interface named **gre_interface1**:

```
(config)> add network interface gre_interface1
(config network interface gre_interface1)>
```

3. Set the zone to **internal**:

```
(config network interface gre_interface1)> zone internal
(config network interface gre_interface1)>
```

4. Set the device to the GRE tunnel created in Task three (**/vpn/iptunnel/gre_tunnel1**):

```
(config network interface gre_interface1)> device /vpn/iptunnel/gre_tunnel1
(config network interface gre_interface1)>
```

5. Set **172.31.0.1/30** as the virtual IP address on the GRE tunnel:

```
(config network interface gre_interface1)> ipv4 address 172.31.0.1/30
(config network interface gre_interface1)>
```

6. Save the configuration and apply the change:

```
(config network interface gre_interface1)> save
Configuration saved.
>
```

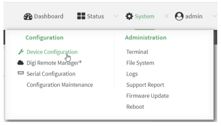
7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Configure the AnywhereUSB Plus-2 device
Task one: Create an IPsec tunnel

WebUI

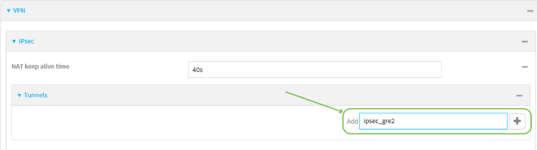
1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.



   The **Configuration** window is displayed.

3. Click **VPN** > **IPsec** > **Tunnels**.

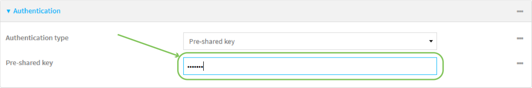
4. For **Add IPsec Tunnel**, type **ipsec_gre2** and click ✚.



NAT keep alive time: 40s

Add: ipsec_gre2

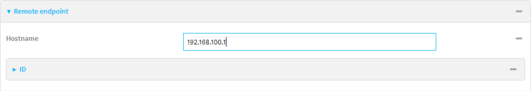
5. Click to expand **Authentication**.
6. For **Pre-shared key**, type the same pre-shared key that was configured for the AnywhereUSB Plus-1 (**testkey**).



Authentication type: Pre-shared key

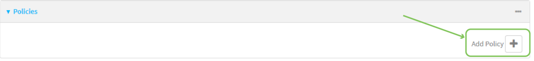
Pre-shared key: •••••••

7. Click to expand **Remote endpoint**.
8. For **Hostname**, type public IP address of the AnywhereUSB Plus-1 device.

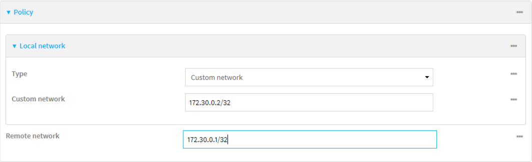


Hostname: 192.168.100.1

9. Click to expand **Policies**.
10. For **Add Policy**, click ✚ to add a new policy.



11. Click to expand **Local network**.
12. For **Type**, select **Custom network**.
13. For **Address**, type the IP address and subnet of the local GRE tunnel, **172.30.0.2/32**.
14. For **Remote network**, type the IP address and subnet of the remote GRE tunnel, **172.30.0.1/32**.

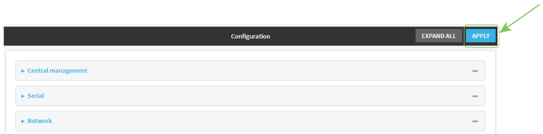


Type: Custom network

Custom network: 172.30.0.2/32

Remote network: 172.30.0.1/32

15. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```
3. Add an IPsec tunnel named **ipsec_gre2**:

   ```
   (config)> add vpn ipsec tunnel ipsec_gre2
   (config vpn ipsec tunnel ipsec_gre2)>
   ```
4. Set the pre-shared key to the same pre-shared key that was configured for the AnywhereUSB Plus-1 (**testkey**):

   ```
   (config vpn ipsec tunnel ipsec_gre2)> auth secret testkey
   (config vpn ipsec tunnel ipsec_gre2)>
   ```
5. Set the remote endpoint to public IP address of the AnywhereUSB Plus-1 device:

   ```
   (config vpn ipsec tunnel ipsec_gre2)> remote hostname 192.168.100.1
   (config vpn ipsec tunnel ipsec_gre2)>
   ```
6. Add a policy:

   ```
   (config vpn ipsec tunnel ipsec_gre2)> add policy end
   (config vpn ipsec tunnel ipsec_gre2 policy 0)>
   ```
7. Set the local network policy type to **custom**:

   ```
   (config vpn ipsec tunnel ipsec_gre2 policy 0)> local type custom
   (config vpn ipsec tunnel ipsec_gre2 policy 0)>
   ```
8. Set the local network address to the IP address and subnet of the local GRE tunnel, **172.30.0.2/32**:

   ```
   (config vpn ipsec tunnel ipsec_gre2 policy 0)> local custom 172.30.0.2/32
   (config vpn ipsec tunnel ipsec_gre2 policy 0)>
   ```
9. Set the remote network address to the IP address and subnet of the remote GRE tunnel, **172.30.0.1/32**:

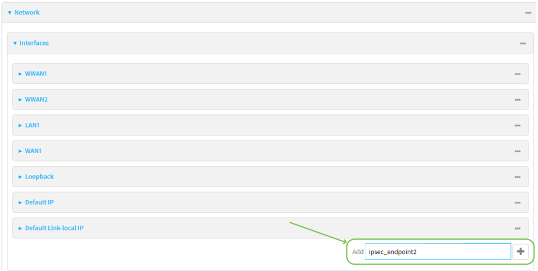
   ```
   (config vpn ipsec tunnel ipsec_gre2 policy 0)> remote network 172.30.0.1/32
   (config vpn ipsec tunnel ipsec_gre2 policy 0)>
   ```
10. Save the configuration and apply the change:

    ```
    (config vpn ipsec tunnel ipsec_gre2 policy 0)> save
    Configuration saved.
    >
    ```

Task two: Create an IPsec endpoint interface

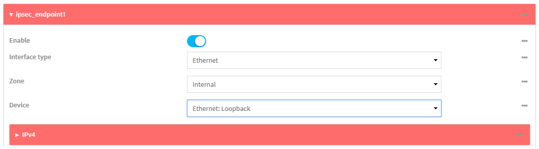
☰ WebUI

1. Click **Network** > **Interfaces**.
2. For **Add Interface**, type **ipsec_endpoint2** and click **+**.



Add ipsec_endpoint2

3. For **Zone**, select **Internal**.
4. For **Device**, select **Ethernet: loopback**.



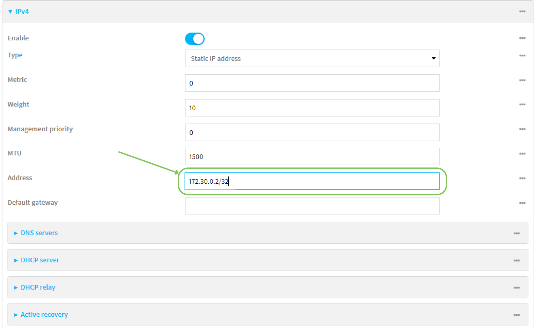
Enable

Interface type Ethernet

Zone Internal

Device Ethernet: Loopback

5. Click to expand **IPv4**.
6. For **Address**, type the IP address of the local GRE tunnel, **172.30.0.2/32**.



Enable

Type Static IP address

Metric 0

Weight 10

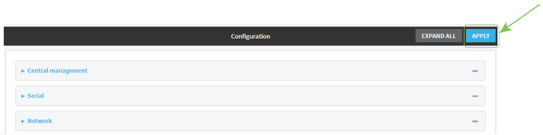
Management priority 0

MTU 1500

Address 172.30.0.2/32

Default gateway

7. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add an interface named **ipsec_endpoint2**:

```
(config)> add network interface ipsec_endpoint2
(config network interface ipsec_endpoint2)>
```

3. Set the zone to **internal**:

```
(config network interface ipsec_endpoint2)> zone internal
(config network interface ipsec_endpoint2)>
```

4. Set the device to **/network/device/loopback**:

```
(config network interface ipsec_endpoint2)> device /network/device/loopback
(config network interface ipsec_endpoint2)>
```

5. Set the IPv4 address to the IP address of the local GRE tunnel, **172.30.0.2/32**:

```
(config network interface ipsec_endpoint2)> ipv4 address 172.30.0.2/32
(config network interface ipsec_endpoint2)>
```

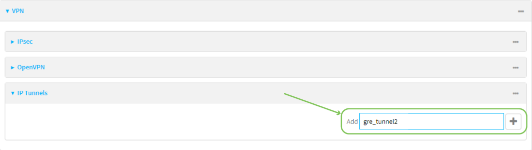
6. Save the configuration and apply the change:

```
(config vpn ipsec tunnel ipsec_endpoint2)> save
Configuration saved.
>
```

Task three: Create a GRE tunnel

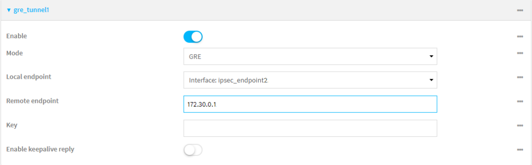
WebUI

1. Click **VPN** > **IP Tunnels**.
2. For **Add IP Tunnel**, type **gre_tunnel2** and click **+**.



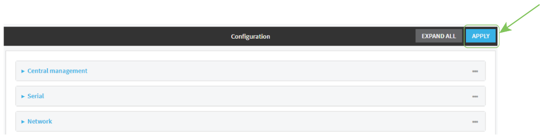
Add gre_tunnel2

3. For **Local endpoint**, select the IPsec endpoint interface created in Task two (**Interface: ipsec_endpoint2**).
4. For **Remote endpoint**, type the IP address of the GRE tunnel on AnywhereUSB Plus-1, **172.30.0.1**.



Enable
Mode: GRE
Local endpoint: Interface: ipsec_endpoint2
Remote endpoint: 172.30.0.1
Key
Enable keepalive reply

5. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add a GRE tunnel named **gre_tunnel2**:

```
(config)> add vpn iptunnel gre_tunnel2
(config vpn iptunnel gre_tunnel2)>
```

3. Set the local endpoint to the IPsec endpoint interface created in Task two (**/network/interface/ipsec_endpoint2**):

```
(config vpn iptunnel gre_tunnel2)> local /network/interface/ipsec_endpoint2
(config vpn iptunnel gre_tunnel2)>
```

4. Set the remote endpoint to the IP address of the GRE tunnel on AnywhereUSB Plus-1, **172.30.0.1**:

```
(config vpn iptunnel gre_tunnel2)> remote 172.30.0.1
(config vpn iptunnel gre_tunnel2)>
```

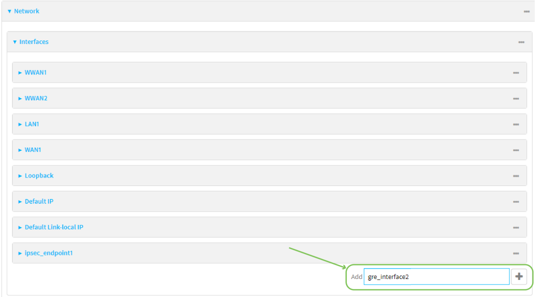
5. Save the configuration and apply the change:

```
(config vpn iptunnel gre_tunnel2)> save
Configuration saved.
>
```

Task four: Create an interface for the GRE tunnel device

WebUI

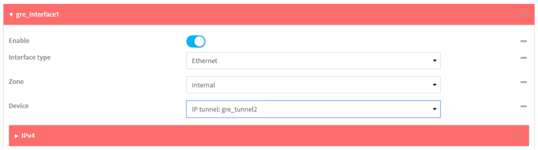
1. Click **Network** > **Interfaces**.
2. For **Add Interface**, type **gre_interface2** and click **+**.



gre_interface2

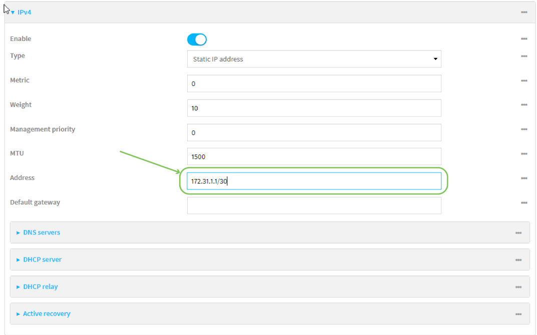
3. For **Zone**, select **Internal**.

4. For **Device**, select the GRE tunnel created in Task three (**IP tunnel: gre_tunnel2**).



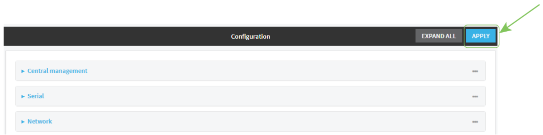
Enable: on
Interface type: Ethernet
Zone: Internal
Device: IP tunnel: gre_tunnel2

5. Click to expand **IPv4**.
6. For **Address**, type **172.31.1.1/30** for a virtual IP address on the GRE tunnel.



Enable: on
Type: Static IP address
Metric: 0
Weight: 10
Management priority: 0
MTU: 1500
Address: 172.31.1.1/30
Default gateway:

7. Click **Apply** to save the configuration and apply the change.



Command line

1. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

2. Add an interface named **gre_interface2**:

```
(config)> add network interface gre_interface2
(config network interface gre_interface2)>
```

3. Set the zone to **internal**:

```
(config network interface gre_interface2)> zone internal
(config network interface gre_interface2)>
```

4. Set the device to the GRE tunnel created in Task three (**/vpn/iptunnel/gre_tunnel2**):

```
(config network interface gre_interface2)> device /vpn/iptunnel/gre_tunnel2
(config network interface gre_interface2)>
```

5. Set **172.31.0.1/30** as the virtual IP address on the GRE tunnel:

```
(config network interface gre_interface2)> ipv4 address 172.31.1.1/30
(config network interface gre_interface2)>
```

6. Save the configuration and apply the change:

```
(config network interface gre_interface2)> save
Configuration saved.
>
```

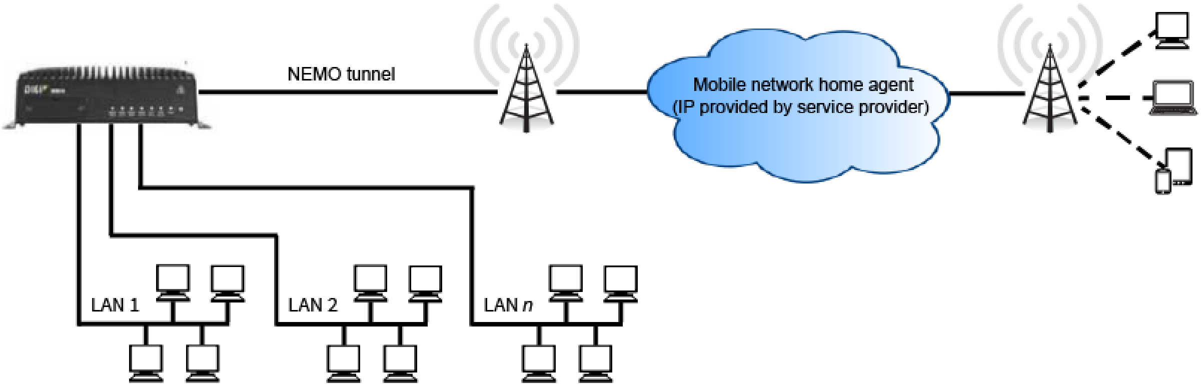
7. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

NEMO

Network Mobility (NEMO) is a mobile networking technology that provides access to one or more Local Area Networks (LANs) on your device. NEMO creates a tunnel between the home agent on the mobile private network and the AnywhereUSB Plus device, isolating the connection from internet traffic and advertising the IP subnets of the LANs for remote access and device management.

Dynamic Mobile Network Routing (DMNR) is the implementation of NEMO for Verizon Wireless Private Networks. DMNR support requires the use of Verizon SIM cards that have DMNR enabled.



Configure a NEMO tunnel

Configuring an NEMO tunnel with a remote device involves configuring the following items:

Required configuration items

- Enable the NEMO tunnel.

  The NEMO tunnel is enabled by default.
- The IP address of the NEMO virtual network interface.
- The firewall zone of the NEMO tunnel.
- The IP address of the NEMO home agent server. This is provided by your cellular carrier.
- The home agent's authentication key. This is provided by your cellular carrier.

- Home agent registration lifetime. This is provided by your cellular carrier.
- The local network interfaces that will be advertised on NEMO.

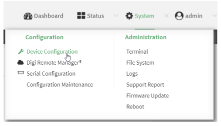
Additional configuration items

- The home agent Software Parameter Index (SPI).
- Path MTU discovery.

  Path MTU discovery is enabled by default. If it is disabled, identify the MTU.
- Care of address: the local network interface that is used to communicate with the peer.
  - If set to **Interface**, identify the local interface to be used. Generally, this will be the Wirelesss WAN (**Modem**).
  - If set to **IP address**, enter the IP address.
- The local network of the GRE endpoint negotiated by NEMO.
  - If the local network is set to Interface, identify the local interface to be used.

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with full Admin access rights.
2. On the menu, click **System**. Under **Configuration**, click **Device Configuration**.

   The **Configuration** window is displayed.
3. Click **VPN** > **NEMO**.

   The NEMO tunnel is enabled by default. To disable, click to toggle off **Enable**.
4. For **Home IP address**, type the IPv4 address of the NEMO virtual network interface.
5. For **Zone**, select the firewall zone for the NEMO tunnel.
6. For **Home agent server IP** address, type the IPv4 address of the NEMO home agent. This is provided by your cellular carrier.
7. For **Key**, type the key used to authenticate to the home agent. This is provided by your cellular carrier.
8. For **Home agent SPI**, type the Security Parameter Index (SPI) value, which is used in the authentication extension when registering. This should be normally left at the default setting of **256** unless your service provider indicates a different value.
9. For **Home agent registration lifetime, in seconds**, type the number of seconds number of seconds until the authorization key expires. This is provided by your cellular carrier.
10. For **MTU discovery**, leave enabled to determine the maximum transmission unit (MTU) size.

    If disabled, for **MTU**, type the MTU size. The default MTU size for LANs on the AnywhereUSB Plus device is 1500. The MTU size of the NEMO tunnel will be smaller, to take into account the required headers.
11. Click to expand **Care of address** to configure the local WAN interface of the internet facing network.

a. For **Type**, select the method to determine the local network interface that is used to communicate with the peer.

- If **Default route** is selected, the network interface that is used will be the same as the default route.
- If **Interface** is selected, specify the local network interface.
- If **IP address** is selected, type the IP address.

The default is **Default route**.

12. Click to expand **GRE tunnel local endpoint**.

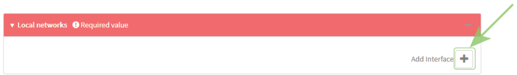
a. For **Type**, select the local endpoint of the GRE endpoint negotiated by NEMO.

- If **Default route** is selected, the network interface that is used will be the same as the default route.
- If **Interface** is selected, specify the local network interface.

The default is **Default route**.

13. Click to expand **Local networks**.

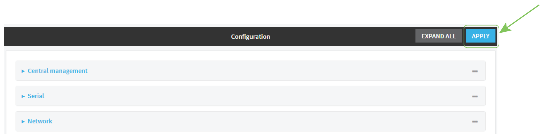
a. For **Add Interface**, click ✚ to add a local network to use as a virtual NEMO network interface.



b. For **Interface**, select the local interface to use as a virtual NEMO network interface. Generally, this will be the a Local Area Network (LAN).

c. (Optional) Repeat for additional interfaces.

14. Click **Apply** to save the configuration and apply the change.



Command line

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

```
> config
(config)>
```

3. Add a NEMO tunnel. For example, to add a NEMO tunnel named **nemo_example**:

```
(config)> add vpn nemo nemo_example
(config vpn nemo nemo_example)>
```

The NEMO tunnel is enabled by default. To disable:

```
(config vpn nemo nemo_example)> enable false
(config vpn nemo nemo_example)>
```

4. Set the IPv4 address of the NEMO virtual network interface:

```
(config vpn nemo nemo_example)> home_address IPv4_address
(config vpn nemo nemo_example)>
```

5. Set the IPv4 address of the NEMO home agent. This is provided by your cellular carrier.

```
(config vpn nemo nemo_example)> home_agent IPv4_address
(config vpn nemo nemo_example)>
```

6. Set the key used to authenticate to the home agent. This is provided by your cellular carrier.

```
(config vpn nemo nemo_example)> key value
(config vpn nemo nemo_example)>
```

7. Set the the number of seconds number of seconds until the authorization key expires. This is provided by your cellular carrier.

```
(config vpn nemo nemo_example)> lifetime integer
(config vpn nemo nemo_example)>
```

   Allowed values are any integer between 1 and 65535.

8. MTU discovery is enabled by default, which allows the device to determine the maximum transmission unit (MTU) size. To disable:

```
(config vpn nemo nemo_example)> mtu_discovery false
(config vpn nemo nemo_example)>
```

   If disabled, set the MTU size. The default MTU size for LANs on the AnywhereUSB Plus device is 1500. The MTU size of the NEMO tunnel will be smaller, to take into account the required headers.

```
(config vpn nemo nemo_example)> mtu integer
(config vpn nemo nemo_example)>
```

   Allowed values are any integer between 68 and 1476.

9. Set the Security Parameter Index (SPI) value, which is used in the authentication extension when registering. This should be normally left at the default setting of **256** unless your service provider indicates a different value.

```
(config vpn nemo nemo_example)> spi integer
(config vpn nemo nemo_example)>
```

   Allowed values are any integer between 256 and 4294967295.

10. Set the firewall zone for the NEMO tunnel:

    ```
    (config vpn nemo nemo_example)> zone zone
    (config vpn nemo nemo_example)>
    ```

    To view a list of available zones:

    ```
    (config vpn nemo nemo_example)> zone ?

    Zone: The firewall zone assigned to this network interface. This can be
    used by
    packet filtering rules and access control lists to restrict network traffic
    on
    this interface.
    Format:
      any
      dynamic_routes
      edge
      external
      internal
      ipsec
      loopback
      setup
    Current value:

    (config vpn nemo nemo_example)> zone
    ```

11. Configure the Care-of-Address, the local WAN interface of the internet facing network.
    a. Set the method to determine the Care-of-Address:

       ```
       (config vpn nemo nemo_example)> coaddress type value
       (config vpn nemo nemo_example)>
       ```

       where *value* is one of:

       - **defaultroute**: Uses the same network interface as the default route.
       - **interface**

         If **interface** is used, set the interface:

         i. Use the **?** to determine available interfaces:

            ```
            (config vpn nemo nemo_example)> coaddress interface ?

            Interface: Use the IP address of this network interface as
            this node's Care-of-Address.
            Format:
              defaultip
              defaultlinklocal
              eth1
              eth2
              loopback
            Current value:
            ```

```
(config vpn nemo nemo_example)> coaddress interface
```

ii. Set the interface. For example:

```
(config vpn nemo nemo_example)> coaddress interface eth1
(config vpn nemo nemo_example)>
```

- **ip**

  If **ip** is used, set the IP address:

  ```
  (config vpn nemo nemo_example)> coaddress address IP_address
  (config vpn nemo nemo_example)>
  ```

The default is **defaultroute**.

12. Set the GRE tunnel local endpoint:

    a. Set the method to determine the GRE tunnel local endpoint:

    ```
    (config vpn nemo nemo_example)> tun_local type value
    (config vpn nemo nemo_example)>
    ```

    where *value* is one of:

    - **defaultroute**: Uses the same network interface as the default route.
    - **interface**

      If **interface** is used, set the interface.

      i. Use the **?** to determine available interfaces:

      ```
      (config vpn nemo nemo_example)> tun_local interface ?

      Interface: The network interface to use to communicate with
      the peer. Set this field to blank if using the default route.
      Format:
        defaultip
        defaultlinklocal
        eth1
        eth2
        loopback
      Current value:

      (config vpn nemo nemo_example)> tun_local interface
      ```

      ii. Set the interface. For example:

      ```
      (config vpn nemo nemo_example)> tun_local interface eth1
      (config vpn nemo nemo_example)>
      ```

    The default is **defaultroute**.

13. Configure one or more local networks to use as a virtual NEMO network interface. Generally, this will be a Local Area Network (LAN):

a. Add a local network to use as a virtual NEMO network interface:

```
(config vpn nemo nemo_example)> add network end eth2
(config vpn nemo nemo_example)>
```

b. (Optional) Repeat for additional interfaces.

14. Save the configuration and apply the change:

```
(config)> save
Configuration saved.
>
```

15. Type **exit** to exit the Admin CLI.

    Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Show NEMO status

WebUI

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. On the menu, select **Status** > **NEMO**.

   The **NEMO** page appears.
3. To view configuration details about an NEMO tunnel, click the 🔧 (configuration) icon in the upper right of the tunnel's status pane.

Command line

1. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.
2. To display details about all configured NEMO tunnel, type the following at the prompt:

```
> show nemo

 NEMO  Enable  Status  Address  Agent    CoAddress
 ----  ------  ------  -------  -------  ----------
 demo  false
 test  true    up      1.2.3.4  4.3.2.1  10.10.10.1

>
```

3. To display details about a specific tunnel:

```
> show nemo name test

 test NEMO Status
 ----------------
 Enabled                  : true
 Status                    : up
 Home Agent                : 4.3.2.1
 Care of Address           : 10.10.10.1
 Interface                 : modem
 GRE Tunnel                : 10.10.10.1 === 4.3.2.1
 Metric                    : 255
 MTU                       : 1476
 Lifetime (Actual)         : 600

 Local Network  Subnet          Status
 -------------  --------------  ----------
 lan1           192.168.2.1/24  Advertized
 LAN2           192.168.3.1/24  Advertized


>
```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Command line interface

This chapter contains the following topics:

Access the command line interface

You can access the AnywhereUSB Plus command line interface using an SSH connection or a serial connection. You can use an open-source terminal software, such as PuTTY or TeraTerm, to access the device through one of these mechanisms.

You can also access the command line interface in the WebUI by using the **Terminal**, or the Digi Remote Manager by using the **Console**.

To access the command line, your device must be configured to allow access, and you must log in as a user who has been configured for the appropriate access. For further information about configuring access to these services, see:

- WebUI: Configure the web administration service
- SSH: Configure SSH access

Log in to the command line interface

Command line

1. Connect to the AnywhereUSB Plus device by using a serial connection, SSH, or the **Terminal** in the WebUI or the **Console** in the Digi Remote Manager. See Access the command line interface for more information.
    - For serial connections, the default configuration is:
        - **115200** baud rate
        - **8** data bits
        - **no** parity
        - **1** stop bit
        - **no** flow control
    - For SSH connections, enter the device's default IP address.
2. At the login prompt, enter the username and password of a user with Admin access:

    ```
    login: admin
    Password: **********
    ```

    The default username is **admin**. The default unique password for your device is printed on the device label.
3. Depending on the device configuration, you may be presented with another menu, for example:

    ```
    Access selection menu:

           a: Admin CLI
           1: Serial: port1        (9600,8,1,none,none)
           q: Quit

    Select access or quit [admin] :
    ```

    Type **a** or **admin** to access the AnywhereUSB Plus command line.

You will now be connected to the Admin CLI:

```
Connecting now, 'exit' to disconnect from Admin CLI ...

>
```

See Command line interface for detailed instructions on using the command line interface.

Exit the command line interface

Command line

1. At the command prompt, type **exit**.

   ```
   > exit
   ```

2. Depending on the device configuration, you may be presented with another menu, for example:

   ```
   Access selection menu:

          a: Admin CLI
          1: Serial: port1        (9600,8,1,none,none)
          q: Quit

   Select access or quit [admin] :
   ```

   Type **q** or **quit** to exit.

Execute a command from the web interface

1. Log into the AnywhereUSB Plus WebUI as a user with Admin access.
2. At the main menu, click **Terminal**. The device console appears.

   ```
   AnywhereUSB Plus login:
   ```

3. Log into the AnywhereUSB Plus command line as a user with Admin access.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

   The Admin CLI prompt appears.

   ```
   >
   ```

Display help for commands and parameters

The help command

When executed from the root command prompt, **help** displays information about autocomplete operations, how to move the cursor on the AnywhereUSB Plus command line, and other keyboard shortcuts:

```
> help

 Commands
 -------------------------------------------------------------------------------
 ?             Show commands help
 <Tab>         Tab completion, displays all valid commands to complete command,
               if only one command is possible, it is used
 <Space>       Like tab except shortest prefix is used if command is valid
 <Enter>       Enter an input. If quoting then a new line is created instead. If
               the input is invalid then characters will be deleted until a
               prefix for a valid command is found.
 Ctrl + A      Move cursor to start of line
 Ctrl + E      Move cursor to end of line
 Ctrl + W      Delete word under cursor until start of line or [\',", ,\,/,.]
 Ctrl + R      If the current input is invalid then characters will be deleted
               until a prefix for a valid command is found.
 Ctrl + left   Jump cursor left until start of line or [\',", ,\,/,.]
 Ctrl + right  Jump cursor right until start of line or [\',", ,\,/,.]

>
```

The question mark (?) command

When executed from the root command prompt, **?** displays available commands:

```
> ?

 Commands
 -------------------------------------------------------------------------------
 config      View and modify the configuration
 exit        Exit the CLI
 analyzer    Analyzer commands.
 cp          Copy a file or directory.
 help        Show CLI editing and navigation commands.
 ls          List a directory.
 mkdir       Create a directory.
 modem       Modem commands.
 more        View a file.
 mv          Move a file or directory.
 ping        Ping a host.
 reboot      Reboot the system.
 rm          Remove a file or directory.
 scp         Copy a file or directory over SSH.
 show        Show instance statistics.
 system      System commands.

 traceroute  Print the route packets trace to network host.
 update      Update firmware.

>
```

Display help for individual commands

When included with a command name, both **?** and **help** provide further information about the command. For example:

1. To display further information about the **show** command, type either **show ?** or **show help**:

```
> show ?

 Commands
 --------------------------------------------------------------------------

 arp          Show ARP tables
 cloud        Show drm statistics
 config       Show config deltas.
 dhcp-lease   Show DHCP leases.
 event        Show event list
 ipsec        Show IPsec statistics.
 log          Show syslog.
 manufacture  Show manufacturer information. modem        Show modem
statistics.
 network      Show network interface statistics.
 openvpn      Show OpenVPN statistics.
 route        Show IP routing information.
  serial       Show serial statistics.
 system       Show system statistics.
 version      Show firmware version.


 > show
```

Use the Tab key or the space bar to display abbreviated help

When executed from the root command prompt, pressing the **Tab** key or the space bar displays an abbreviated list of available commands:

Similar behavior is available with any command name:

```
> config network interface <space>
..               ...              defaultip        defaultlinklocal lan
loopback
> config network interface
```

Auto-complete commands and parameters

When entering a command and parameter, press the **Tab** key to cause the command line interface to auto-complete as much of the command and parameter as possible. Typing the space bar has similar behavior. If multiple commands are available that will match the entered text, auto-complete is not performed and the available commands are displayed instead.

Auto-complete applies to these command elements only :

- Command names. For example, typing **net<Tab>** auto-completes the command as **network**.
- Parameter names. For example:
  - **ping *hostname* int<Tab>** auto-completes the parameter as **interface**.
  - **system b<Tab>** auto-completes the parameter as **backup**.
- Parameter values, where the value is one of an enumeration or an on|off type; for example:

  ```
  (config)> serial port1 enable t<Tab>
  ```

  auto-completes to

  ```
  (config)> serial port1 enable true
  ```

Auto-complete does not function for:

- Parameter values that are string types.
- Integer values.
- File names.
- Select parameters passed to commands that perform an action.

Available commands

The following commands are available from the Admin CLI prompt:

Command	Description
config	Used to view and modify the configuration. See Device configuration using the command line interface for more information about using the **config** command.
exit	Exits the CLI.
cp	Copies a file or directory.
help	Displays: ■ CLI editing and navigation commands, when executed from the root of the Admin CLI prompt. ■ Available commands, syntax diagram, and parameter information, when executed in conjunction with another command. See Display help for commands and parameters for information about the **help** command.
ls	Lists the contents of a directory.
mkdir	Creates a directory.
modem	Executes modem commands.
more	Displays the contents of a file.
mv	Moves a file or directory.
ping	Pings a remote host using Internet Control Message Protocol (ICMP) Echo Request messages.
reboot	Reboots the AnywhereUSB Plus device.
rm	Removes a file.
scp	Uses the secure copy protocol (SCP) to transfer files between the AnywhereUSB Plus device and a remote host. See Use the scp command for information about using the **scp** command.
show	Displays information about the device and the device's configuration. See Display status and statistics using the show command for more information about the show command.
system	Issues commands related to system functionality.
traceroute	Sends and tracks route packets to a destination host.
update	Updates the device firmware.

Note For commands that operate on the AnywhereUSB Plus's file system, such as the **cp**, **ls**, and **mkdir** commands, see File system for information about the file system, including how to copy, move and delete files and directories.

Use the scp command

The **scp** command uses Secure Copy Protocol (SCP) to transfer files between the AnywhereUSB Plus device and a remote host.

Required configuration items

- The hostname or IP address of the remote host.
- The username and password of the user on the remote host.
- Whether the file is being copied to the AnywhereUSB Plus device from a remote host, or to the remote host from the AnywhereUSB Plus device.
    - If the file is being copied to the AnywhereUSB Plus device from a remote host:
        - The path and filename of the file on the remote host that will be copied to the AnywhereUSB Plus device.
        - The location on the AnywhereUSB Plus device where the file will be copied.
    - If the file is being copied to a remote host from the AnywhereUSB Plus device:
        - The path and filename of the file on the AnywhereUSB Plus device that will be copied to the remote host.
        - The location on the remote host where the file will be copied.

Copy a file from a remote host to the AnywhereUSB Plus device

To copy a file from a remote host to the AnywhereUSB Plus device, use the scp command as follows:

```
> scp host hostname-or-ip user username remote remote-path local local-path to
local
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the path and filename of the file on the remote host that will be copied to the AnywhereUSB Plus device.
- *local-path* is the location on the AnywhereUSB Plus device where the copied file will be placed.

Transfer a file from the AnywhereUSB Plus device to a remote host

To copy a file from the AnywhereUSB Plus device to a remote host, use the scp command as follows:

```
> scp host hostname-or-ip user username remote remote-path local local-path to
remote
```

where:

- *hostname-or-ip* is the hostname or ip address of the remote host.
- *username* is the name of the user on the remote host.
- *remote-path* is the location on the remote host where the file will be copied.
- *local-path* is the path and filename on the AnywhereUSB Plus device.

To copy a support report from the AnywhereUSB Plus device to a remote host at the IP address of 192.168.4.1:

1. Use the **system support-report** command to generate the report:

```
> system support-report /var/log/
Saving support report to /var/log/support-report-0040D0133536-21-02-26-
8:04:23.bin
Support report saved.
>
```

2. Use the **scp** command to transfer the report to a remote host:

```
> scp host 192.168.4.1 user admin remote /home/admin/temp/ local
/var/log/support-report-00:40:D0:13:35:36-21-02-26-8:04:23.bin to remote
admin@192.168.4.1's password: adminpwd
support-report-0040D0133536-21-02-26-8:04:23.bin
>
```

Display status and statistics using the show command

The AnywhereUSB Plus **show** command display status and statistics for various features.

For example:

show config

The show config command displays all the configuration settings for the device that have been changed from the default settings. This is a particularly useful when troubleshooting the device.

```
> show config

auth tacacs+ service "login"
auth user admin password
"$2a$05$WlJQhquI7BgsytkpobKhaeLPtWraGANBcrlEaJX/wJv63JENW/HOu"
add auth user test
add auth user test group end "admin"
add auth user test group end "serial"
auth user test password
"$2a$05$RdGYz1sLKbWrqe6cZjlsd.otg03JZR6n9939XV6EYWUSP0tMAzO5W"
network interface lan ipv4 type "dhcp"
network interface lan zone "external"
network interface modem modem apn 0 apn "00000.000"
network interface modem modem apn_lock "true"
schema version "445"

>
```

show system

The show system command displays system information and statistics for the device, including CPU usage.

```
> show system

  Model                    : Digi AnywhereUSB Plus
  Serial Number            : AnywhereUSB Plus-000065
  SKU                      : AnywhereUSB Plus
  Hostname                 : AnywhereUSB Plus
  MAC                      : DF:DD:E2:AE:21:18

  Hardware Version         : 50001947-01 1P
  Firmware Version         : 21.2.39.67
  Alt. Firmware Version    : 21.2.39.67
  Bootloader Version       : 19.7.23.0-15f936e0ed

  Current Time             : Fri, 26 Feb 2021 8:04:23 +0000
  CPU                      : 1.4%
  Uptime                   : 6 days, 6 hours, 21 minutes, 57 seconds (541317s)
  Temperature              : 40C

>
```

show network

The show network command displays status and statistics for network interfaces.

```
> show network

 Interface          Proto  Status   Address
 ----------------  -----  -------  -------------------------------
 defaultip          IPv4   up       192.168.210.1/24
 defaultlinklocal   IPv4   up       169.254.100.100/16
 lan              IPv4   up       192.168.2.1
 lan              IPv6   up       0:0:0:0:0:ffff:c0a8:301
 loopback           IPv4   up       127.0.0.1/8
 wan              IPv4   up       192.168.3.1/24
 wan              IPv6   up       fd00:2704::240:ffff:fe80:120/64

>
```

Device configuration using the command line interface

The **config** command allows for device configuration from the command line. All configuration tasks that can be performed by using the WebUI can also be performed by using the **config** command.

There are two ways to invoke the **config** command from the CLI:

- Execute the **config** command and parameters at the root prompt. See Execute configuration commands at the root Admin CLI prompt for more information.
- Enter configuration mode by executing the **config** command without any parameters. See Configuration mode for more information.

Execute configuration commands at the root Admin CLI prompt

You can execute the **config** command at the root Admin CLI prompt with any appropriate parameters. When the **config** command is used in this way, changes to the device's configuration are automatically saved when the command is executed.

For example, to disable the SSH service from the root prompt, enter the following command:

```
> config service ssh enable false
>
```

The AnywhereUSB Plus device's ssh service is now disabled.

Note When the **config** command is executed at the root prompt, certain configuration actions that are available in configuration mode cannot be performed. This includes validating configuration changes, canceling and reverting configuration changes, and performing actions on elements in lists. See Configuration mode for information about using configuration mode.

Display help for the config command from the root Admin CLI prompt

Display additional configuration commands, as well as available parameters and values, by entering the question mark (?) character after the **config** command.

1. For example:

   ```
   > config ?
   ```

   Will display the following help information:

   ```
   > config ?

   Additional Configuration
   --------------------------------------------------------------------------
   application               Custom scripts
   auth                      Authentication
   cloud                     Central management
   firewall                  Firewall
   monitoring                Monitoring
   network                   Network
   serial                    Serial
   service                   Services
   system                    System
   vpn                       VPN

   Run "config" with no arguments to enter the configuration editing mode.

   > config
   ```

2. You can then display help for the additional configuration commands. For example, to display help for the **config service** command:

```
> config service ?
Services

Additional Configuration
--------------------------------------------------------------------------
dns                       DNS
mdns                      Service Discovery (mDNS)
multicast                 Multicast
ntp                       NTP
remote_control            Remote control
snmp                      SNMP
ssh                       SSH
web_admin                 Web administration

> config service
```

3. Next, display help for the **config service ssh** command:

```
> config service ssh ?

SSH: An SSH server for managing the device.

Parameters                Current Value
--------------------------------------------------------------------------
enable                    true           Enable
key                       [private]      Private key
port                      22             Port

Additional Configuration
--------------------------------------------------------------------------
acl                       Access control list
mdns

> config service ssh
```

4. Lastly, display the allowed values and other information for the **enable** parameter:

```
> config service ssh enable ?

Enable: Enable the service.
Format: true, false, yes, no, 1, 0
Default value: true
Current value: true

> config service ssh enable
```

Configuration mode

Configuration mode allows you to perform multiple configuration tasks and validate the changes prior to saving them. You can cancel all changes without saving them at any time. Configuration changes do not take effect until the configuration is saved.

Enable configuration mode

To enable configuration mode, at the root prompt, enter the **config** command without any parameters:

```
> config
(config)>
```

When the command line is in configuration mode, the prompt will change to include **(config)**, to indicate that you are currently in configuration mode.

Enter configuration commands in configuration mode

There are two ways to enter configuration commands while in configuration mode:

- Enter the full command string from the config prompt.

  For example, to disable the ssh service by entering the full command string at the config prompt:

  ```
  (config)> service ssh enable false
  (config)>
  ```

- Execute commands by moving through the configuration schema.

  For example, to disable the ssh service by moving through the configuration and then executing the **enable false** command:

  1. At the **config** prompt, enter **service** to move to the **service** node:

     ```
     (config)> service
     (config service)>
     ```

  2. Enter **ssh** to move to the **ssh** node:

     ```
     (config service)> ssh
     (config service ssh)>
     ```

  3. Enter **enable false** to disable the **ssh** service:

     ```
     (config service ssh)> enable false
     (config service ssh)>
     ```

  See Move within the configuration schema for more information about moving within the configuration.

Save changes and exit configuration mode

To save changes that you have made to the configuration while in configuration mode, use **save**. The save command automatically validates the configuration changes; the configuration will not be saved if it is not valid. Note that you can also validate configuration changes at any time while in configuration mode by using the **validate** command.

```
(config)> save
Configuration saved.
>
```

After using **save** to save changes to the configuration, you will automatically exit configuration mode. To return to configuration mode, type **config** again.

Exit configuration mode without saving changes

You can discard any unsaved configuration changes and exit configuration mode by using the **cancel** command:

```
(config)> cancel
>
```

After using **cancel** to discard unsaved changes to the configuration, you will automatically exit configuration mode.

Configuration actions

In configuration mode, configuration actions are available to perform tasks related to saving or canceling the configuration changes, and to manage items and elements in lists. The commands can be listed by entering a question mark (**?**) at the **config** prompt.

The following actions are available:

Configuration actions	Description
cancel	Discards unsaved configuration changes and exits configuration mode.
save	Saves configuration changes and exits configuration mode.
validate	Validates configuration changes.
revert	Reverts the configuration to default settings. See The revert command for more information.
show	Displays configuration settings.
add	Adds a named element, or an element in a list. See Manage elements in lists for information about using the **add** command with lists.
del	Deletes a named element, or an element in a list. See Manage elements in lists for information about using the **del** command with lists.
move	Moves elements in a list. See Manage elements in lists for information about using the **move** command with lists.

Display command line help in configuration mode

Display additional configuration commands, as well as available parameters and values, by entering the question mark (?) character at the **config** prompt. For example:

1. Enter **?** at the **config** prompt:

   ```
   (config)> ?
   ```

   This will display the following help information:

   ```
   (config)> ?

    Additional Configuration
    --------------------------------------------------------------------------
    application               Custom scripts
    auth                      Authentication
    cloud                     Central management
    firewall                  Firewall
    monitoring                Monitoring
    network                   Network
    serial                    Serial
    service                   Services
    system                    System
    vpn                       VPN

   (config)>
   ```

2. You can then display help for the additional configuration commands. For example, to display help for the **config service** command, use one of the following methods:

   - At the **config** prompt, enter **service ?**:

     ```
     (config)> service ?
     ```

   - At the **config** prompt:

     a. Enter **service** to move to the **service** node:

     ```
     (config)> service
     (config service)>
     ```

     b. Enter **?** to display help for the **service** node:

     ```
     (config service)> ?
     ```

   Either of these methods will display the following information:

   ```
   config> service ?

   Services

    Additional Configuration
    --------------------------------------------------------------------------
    dns                       DNS
    mdns                      Service Discovery (mDNS)
    multicast                 Multicast
    ntp                       NTP
    remote_control            Remote control
   ```

```
  snmp                      SNMP
  ssh                       SSH
  web_admin                 Web administration

 (config)> service
```

3. Next, to display help for the **service ssh** command, use one of the following methods:
   - At the **config** prompt, enter **service ssh ?**:

     ```
     (config)> service ssh ?
     ```

   - At the **config** prompt:
     a. Enter **service** to move to the **service** node:

        ```
        (config)> service
        (config service)>
        ```

     b. Enter **ssh** to move to the **ssh** node:

        ```
        (config service)> ssh
        (config service ssh)>
        ```

     c. Enter **?** to display help for the **ssh** node:

        ```
        (config service ssh)> ?
        ```

   Either of these methods will display the following information:

```
 (config)> service ssh ?

 SSH: An SSH server for managing the device.

  Parameters                Current Value
  --------------------------------------------------------------------------
  enable                    true           Enable
  key                       [private]      Private key
  port                      22             Port

  Additional Configuration
  --------------------------------------------------------------------------
  acl                       Access control list
  mdns

 (config)> service ssh
```

4. Lastly, to display allowed values and other information for the **enable** parameter, use one of the following methods:
   - At the **config** prompt, enter **service ssh enable ?**:

     ```
     (config)> service ssh enable ?
     ```

- At the **config** prompt:
  a. Enter **service** to move to the **service** node:

     ```
     (config)> service
     (config service)>
     ```

  b. Enter **ssh** to move to the **ssh** node:

     ```
     (config service)> ssh
     (config service ssh)>
     ```

  c. Enter **enable ?** to display help for the **enable** parameter:

     ```
     (config service ssh)> enable ?
     (config service ssh)>
     ```

Either of these methods will display the following information:

```
(config)> service ssh enable ?

 Enable: Enable the service.
 Format: true, false, yes, no, 1, 0
 Default value: true
 Current value: true

(config)> service ssh enable
```

Move within the configuration schema

You can perform configuration tasks at the CLI by moving within the configuration.

- Move forward one node in the configuration by entering the name of an Additional Configuration option:
  1. At the **config** prompt, type **service** to move to the **service** node:

     ```
     (config)> service
     (config service)>
     ```

  2. Type **ssh** to move to the **ssh** node:

     ```
     (config service)> ssh
     (config service ssh)>
     ```

  3. Type **acl** to move to the **acl** node:

     ```
     (config service ssh)> acl
     (config service ssh acl)>
     ```

  4. Type **zone** to move to the **zone** node:

     ```
     (config service ssh acl)> zone
     (config service ssh acl zone)>
     ```

You can also enter multiple nodes at once to move multiple steps in the configuration:

```
(config)> service ssh acl zone
(config service ssh acl zone)>
```

- Move backward one node in the configuration by entering two periods (**..**):

```
(config service ssh acl zone)> ..
(config service ssh acl)>
```

You can also move back multiples nodes in the configuration by typing multiple sets of two periods:

```
(config service ssh acl zone)> .. .. ..
(config service)>
```

- Move to the root of the config prompt from anywhere within the configuration by entering three periods (**...**):

```
(config service ssh acl zone)> ...
(config)>
```

Manage elements in lists

While in configuration mode, you can use the **add**, **del**, and **move** action commands to manage elements in a list. When working with lists, these actions require an index number to identify the list item that will be acted on.

Add elements to a list

When used with parameters that contains lists of elements, the **add** command is used to add an element to the list.

For example, to add an authentication method:

1. Display current authentication method by using the **show** command:

```
(config)> show auth method
0 local
(config)>
```

2. Add an authentication method by using the **add *index_item*** command. For example:
    - To add the TACACS+ authentication method to the beginning of the list, use the index number **0**:

```
(config)> add auth method 0 tacacs+
(config)> show auth method
0 tacacs+
1 local
(config)>
```

- To add the TACACS+ authentication method to the end of the list, use the **end** keyword:

```
(config)> add auth method end tacacs+
(config)> show auth method
0 local
1 tacacs+
(config)>
```

The end keyword

As demonstrated above, the **end** keyword is used to add an element to the end of a list. Additionally, the **end** keyword is used to add an element to a list that does not have any elements.

For example, to add an authentication group to a user that has just been created:

1. Use the **show** command to verify that the user is not currently a member of any groups:

```
(config)> show auth user new-user group
(config)>
```

2. Use the **end** keyword to add the admin group to the user's configuration:

```
(config)> add auth user new-user group end admin
(config)>
```

3. Use the **show** command again to verify that the admin group has been added to the user's configuration:

```
(config)> show auth user new-user group
0 admin
(config)>
```

Delete elements from a list

When used with parameters that contains lists of elements, the **del** command is used to delete an element in the list.

For example, to delete an authentication method:

1. Use the **show** command to display current authentication method configuration:

```
(config)> show auth method
0 local
1 tacacs+
2 radius
(config)>
```

2. Delete one of the authentication methods by using the **del** ***index_number*** command. For example:
   a. To delete the local authentication method, use the index number **0**:

```
(config)> del auth method 0
(config)>
```

   b. Use the **show** command to verify that the local authentication method was removed:

```
(config)> show auth method
0 tacacs+
1 radius
(config)>
```

Move elements within a list

Use the **move** command to reorder elements in a list.

For example, to reorder the authentication methods:

1. Use the **show** command to display current authentication method configuration:

   ```
   (config)> show auth method
   0 local
   1 tacacs+
   2 radius
   (config)>
   ```

2. To configure the device to use TACACS+ authentication first to authenticate a user, use the **move *index_number_1 index_number_2*** command:

   ```
   (config)> move auth method 1 0
   (config)>
   ```

3. Use the **show** command again to verify the change:

   ```
   (config)> show auth method
   0 tacacs+
   1 local
   2 radius
   (config)>
   ```

The revert command

The **revert** command is used to revert changes to the AnywhereUSB Plus device's configuration and restore default configuration settings. The behavior of the revert command varies depending on where in the configuration hierarchy the command is executed, and whether the optional **path** parameter is used. After executing the revert command, you must save the configuration changes by using the **save** command. You can also discard the configuration changes by using the **cancel** command.

CAUTION! The **revert** command reverts all changes to the default configuration, not only unsaved changes.

Revert all configuration changes to default settings

To discard all configuration changes and revert to default settings, use the **revert** command at the config prompt without the optional **path** parameter:

1. At the config prompt, enter **revert**:

   ```
   (config)> revert
   (config)>
   ```

2. Set the password for the admin user prior to saving the changes:

   ```
   (config)> auth user admin password pwd
   (config)>
   ```

3. Save the configuration and apply the change:

   ```
   (config)> save
   Configuration saved.
   >
   ```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Revert a subset of configuration changes to the default settings

There are two methods to revert a subset of configuration changes to the default settings.

- Enter the **revert** command with the **path** parameter. For example, to revert all changes to the authentication methods configuration:
  1. Enter the **revert** command with the **path** set to **auth method**:

     ```
     (config)> revert auth method
     (config)>
     ```

  2. Save the configuration and apply the change:

     ```
     (config)> save
     Configuration saved.
     >
     ```

  3. Type **exit** to exit the Admin CLI.

     Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.
- Move to the location in the configuration and enter the **revert** command without the **path** parameter. For example:
  1. Change to the auth method node:

     ```
     (config)> auth method
     (config auth method)>
     ```

  2. Enter the **revert** command:

     ```
     (config auth method)> revert
     (config auth method)>
     ```

3. Save the configuration and apply the change:

   ```
   (config auth method)> save
   Configuration saved.
   >
   ```

4. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

- You can also use a combination of both of these methods:
  1. Change to the **auth** node:

     ```
     (config)> auth
     (config auth)>
     ```

  2. Enter the **revert** command with the **path** set to **method**:

     ```
     (config auth)> revert method
     (config auth)>
     ```

  3. Save the configuration and apply the change:

     ```
     (config auth)> save
     Configuration saved.
     >
     ```

  4. Type **exit** to exit the Admin CLI.

     Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Enter strings in configuration commands

For string parameters, if the string value contains a space, the value must be enclosed in quotation marks. For example, to assign a descriptive name for the device using the **system** command, enter:

```
(config)> system description "Digi AnywhereUSB Plus"
```

Example: Create a new user by using the command line

In this example, you will use the AnywhereUSB Plus command line to create a new user, provide a password for the user, and assign the user to authentication groups.

1. Log into the AnywhereUSB Plus command line as a user with full Admin access rights.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **admin** to access the Admin CLI.

2. At the command line, type **config** to enter configuration mode:

   ```
   > config
   (config)>
   ```

3. At the config prompt, create a new user with the username **user1**:
    - Method one: Create a user at the root of the config prompt:

      ```
      (config)> add auth user user1
      (config auth user user1)>
      ```

    - Method two: Create a user by moving through the configuration:
        a. At the config prompt, enter **auth** to move to the **auth** node:

        ```
        (config)> auth
        (config auth)>
        ```

        b. Enter **user** to move to the **user** node:

        ```
        (config auth)> user
        (config auth user)>
        ```

        c. Create a new user with the username **user1**:

        ```
        (config auth user)> add user1
        (config auth user user1)>
        ```

4. Configure a password for the user:

    ```
    (config auth user user1)> password pwd1
    (config auth user user1)>
    ```

5. List available authentication groups:

    ```
    (config auth user user1)> show .. .. group

    admin
        acl
            admin
                enable true
            nagios
                enable false
            openvpn
                enable false
                no tunnels
            portal
                enable false
                no portals
            serial
                enable false
                no ports
            shell
                enable false

    serial
        acl
            admin
    ```

```
            enable true
        nagios
            enable false
        openvpn
            enable false
            no tunnels
        portal
            enable false
            no portals
        serial
            enable true
                ports
                    0 port1
        shell
            enable false
(config auth user user1)>
```

6. Add the user to the admin group:

```
(config auth user user1)> add group end admin
(config auth user user1)>
```

7. Save the configuration and apply the change:

```
(config auth user user1)> save
Configuration saved.
>
```

8. Type **exit** to exit the Admin CLI.

   Depending on your device configuration, you may be presented with an **Access selection menu**. Type **quit** to disconnect from the device.

Command line reference

analyzer

Analyzer commands.

analyzer clear name STRING

Clears the traffic captured by the analyzer.

Parameters

name

Name of the capture filter to use.

Syntax: STRING

analyzer save filename STRING name STRING

Saves the current captured traffic to a file.

Parameters

filename

The filename to save captured traffic to. The file will be saved to the device's /etc/config/analyzer directory.

Syntax: STRING

name

Name of the capture filter to use.

Syntax: STRING

analyzer start name STRING

Start a capture session of packets on this devices interfaces.

Parameters

name

Name of the capture filter to use.

Syntax: STRING

analyzer stop name STRING

Stops the traffic capture session.

Parameters

name

Name of the capture filter to use.

Syntax: STRING

cp

cp commands.

[force] SOURCE DESTINATION

Copy a file or directory.

Parameters

source

The source file or directory to copy.

Syntax: STRING

destination

The destination path to copy the source file or directory to.

Syntax: STRING

force

Do not ask to overwrite the destination file if it exists.

Syntax: BOOLEAN

Default: False

Optional: True

help

Show CLI editing and navigation commands.

Parameters

None

ls

Directory listing command.

ls [show-hidden] PATH

List a directory.

Parameters

path

List files and directories under this path.

Syntax: STRING

show-hidden

Show hidden files and directories. Hidden filenames begin with '.'.

Syntax: BOOLEAN

Default: False

Optional: True

mkdir

mkdir PATH

Create a directory. Parent directories are created as needed.

Parameters

path

The directory path to create.

Syntax: STRING

modem

Modem commands.

modem at [imei STRING] [name STRING] CMD

Send an AT command to the modem and display the response.

Parameters

cmd

The AT command string.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem at-interactive [imei STRING] [name STRING]

Start an AT command session on the modem's AT serial port.

Parameters

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem firmware

Commands for interacting with cellular modem firmware. See Update cellular module firmware for further information about using the modem firmware commands.

firmware check [imei *STRING*] [name *STRING*]

Inspect /opt/[MODEM_MODEL]/Custom_Firmware/ directory for new modem firmware file.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

firmware list [imei *STRING*] [name *STRING*]

List modem firmware files found in the /opt/[MODEM_MODEL]/ directory.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

firmware ota

Commands for performing FOTA (firmware-over-the-air) interactions with cellular modem.

ota check [imei STRING] [name STRING]

Query the Digi firmware server for the latest remote modem firmware version.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

ota list [imei STRING] [name STRING]

Query the Digi firmware server for a list of modem firmware versions.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

ota update [imei STRING] [name STRING] [version STRING]

Perform FOTA (firmware-over-the-air) update. The modem will be updated to the latest modem firmware image unless a specific firmware version is specified.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

version

Firmware version name

Optional: True

Type: string

firmware update [imei *STRING*] [name *STRING*] [version *STRING*]

Update modem firmware using local firmware file. The modem will be updated to the firmware specified in the /opt/[MODEM_MODEL]/Custom_Firmware/ directory unless a specific firmware version is specified.

Parameters

imei

The IMEI of the modem to execute this CLI command on

Optional: True

Type: string

name

The configured name of the modem to execute this CLI command on

Optional: True

Ref: /network/modem

Type: string

version

Firmware version name

Optional: True

Type: string

modem pin

PIN commands.

pin change [imei *STRING*] [name *STRING*] OLD-PIN NEW-PIN

Change the SIM's PIN code. Warning: Attempting to use an incorrect PIN code may PUK lock the SIM.

Parameters

old-pin

The SIM's PIN code.

Syntax: STRING

new-pin

The PIN code to change to.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

pin disable [imei *STRING*] [name *STRING*] PIN

Disable the PIN lock on the SIM card that is active in the modem. Warning: Attempting to use an incorrect PIN code may PUK lock the SIM.

Parameters

pin

The SIM's PIN code.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

pin enable [imei *STRING*] [name *STRING*] PIN

Enable the PIN lock on the SIM card that is active in the modem. The SIM card will need to be unlocked before each use. Warning: Attempting to use an incorrect PIN code may PUK lock the SIM.

Parameters

pin

The SIM's PIN code.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

pin status [imei *STRING*] [name *STRING*]

Print the PIN lock status and the number of PIN enable/disable/unlock attempts remaining. The SIM will be PUK locked when there are no remaining retries

Parameters

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

pin unlock [imei *STRING*] [name *STRING*] PIN

Temporarily unlock the SIM card with a PIN code. Set the PIN field in the modem interface's configuration to unlock the SIM card automatically before use. Warning: Attempting to use an incorrect PIN code may PUK lock the SIM.

Parameters

pin

The SIM's PIN code.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem puk

PUK commands.

puk status [imei *STRING*] [name *STRING*]

Print the PUK status and the number of PUK unlock attempts remaining.

Parameters

modem puk status [*imei* STRING] [*name* STRING]

Print the PUK status and the number of PUK unlock attempts remaining.

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

puk unlock [imei *STRING*] [name *STRING*] PUK NEW-PIN

Unlock the SIM with a PUK code from the SIM provider.

Parameters

puk

The SIM's PUK code.

Syntax: STRING

new-pin

The PIN code to change to.

Syntax: STRING

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem reset [imei STRING] [name STRING]

Reset the modem hardware (reboot it). This can be useful if the modem has stopped responding to the network or is behaving inconsistently.

Parameters

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem scan [*imei*STRING] [*name*STRING]

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

modem sim-slot [imei STRING] [name STRING] SLOT

Show or change the modem's active SIM slot. This applies only to modems with multiple SIM slots.

Parameters

slot

The SIM slot to change to.
Syntax: (1|2|show)

imei

The IMEI of the modem to execute this CLI command on.
Syntax: STRING
Optional: True

name

The configured name of the modem to execute this CLI command on.
Syntax: STRING
Optional: True

more

path

The file to view.

Syntax: STRING

mv

Move a file or directory.

mv [force] SOURCE DESTINATION

Parameters

source

The source file or directory to move.

Syntax: STRING

destination

The destination path to move the source file or directory to.

Syntax: STRING

force

Do not ask to overwrite the destination file if it exists.

Syntax: BOOLEAN

Default: False

Optional: True

ping

Ping a host using ICMP echo.

ping [broadcast|ipv6] [count INTEGER] [interface STRING] [size INTEGER] [source STRING] HOST

Parameters

host

The name or address of the remote host to send ICMP ping requests to. If broadcast is enabled, can be the broadcast address.

Syntax: STRING

broadcast

Enable broadcast ping functionality

Syntax: BOOLEAN

Default: False

Optional: True

count

The number of ICMP ping requests to send before terminating.

Syntax: INT

Minimum: 1

Default: 100

interface

The network interface to send ping packets from when the host is reachable over a default route. If not specified, the system's primary default route will be used.

Syntax: STRING

Optional: True

ipv6

If a hostname is defined as the value of the 'host' parameter, use the hosts IPV6 address.

Syntax: BOOLEAN

Default: False

Optional: True

size

The number of bytes sent in the ICMP ping request.

Syntax: INT

Minimum: 0

Default: 56

source

The ping command will send a packet with the source address set to the IP address of this interface, rather than the address of the interface the packet is sent from.

Syntax: STRING

Optional: True

reboot

Reboot the system.

Parameters

None

rm

Remove a file or directory.

rm [force] PATH

Parameters

path

The path to remove.

Syntax: STRING

force

Force the file to be removed without asking.

Syntax: BOOLEAN

Default: False

Optional: True

scp

Copy a file or directory over SSH.

scp host STRING local STRING [port INTEGER] remote STRING to STRING user STRING

Parameters

host

The name or address of the remote host.

Syntax: STRING

local

The file to copy to or from on the local device.

Syntax: STRING

port

The SSH port to use to connect to the remote host.

Syntax: INT

Maximum: 65535

Minimum: 1

Default: 22

remote

The file to copy to or from on the remote host.

Syntax: STRING

to

Copy the file from the local device to the remote host, or from the remote host to the local device.

Syntax: (remote|local)

user

The username to use when connecting to the remote host.

Syntax: STRING

show

Show instance status and statistics.

show analyzer name STRING

Show packets from a specified analyzer capture.

Parameters

name

Name of the capture filter to use.

Syntax: STRING

show arp [ipv4|ipv6|verbose]

Show ARP tables, if no IP version is specifed IPv4 IPV6 will be displayed.

Parameters

ipv4

Display IPv4 routes. If no IP version is specifed IPv4 and IPV6 will be displayed

Syntax: BOOLEAN

Default: False

Optional: True

ipv6

Display IPv6 routes. If no IP version is specifed IPv4 and IPV6 will be displayed

Syntax: BOOLEAN

Default: False

Optional: True

verbose

Display more information (less concise, more detail).

Syntax: BOOLEAN

Default: False

Optional: True

show cloud

Show Digi Remote Manager status and statistics.

Parameters

None

show config

Show changes made to default configuration.

Parameters

None

show dhcp-lease [all|verbose]

Show DHCP leases.

Parameters

all

Show all leases (active and inactive (not in etc/config/dhcp.*lease)).

Syntax: BOOLEAN

Default: False

Optional: True

verbose

Display more information (less concise, more detail).

Syntax: BOOLEAN

Default: False

Optional: True

show event [number INTEGER] [table STRING]

Show event list (high level).

Parameters

number

Number of lines to retrieve from log.

Syntax: INT

Minimum: 1

Default: 20

table

Type of event log to be displayed (status, error, info).

Syntax: (status|error|info)

Optional: True

show hotspot [ip STRING] [name STRING]

Show hotspot statistics.

Parameters

ip

IP address of a specific client, to limit the status display to only this client.

Syntax: STRING

Optional: True

name

The configured instance name of the hotspot.

Syntax: STRING

Optional: True

show ipsec [all] [tunnel STRING]

Show IPsec status statistics.

Parameters

all

Display all tunnels including disabled tunnels.

Syntax: BOOLEAN

Default: False

Optional: True

tunnel

Display more details and config data for a specific IPsec tunnel.

Syntax: STRING

Optional: True

verbose

Display status of one or all tunnels in plain text.

Syntax: BOOLEAN

Default: False

Optional: True

show location [geofence]

Show location information.

Parameters

geofence

Shows the status of any configured geofences.

show log [filter STRING] [number INTEGER]

Show system log (low level).

Parameters

filter

Filters for type of log message displayed (critical, warning, info, debug). Note, filters from the number of messages retrieved not the whole log (this can be very time consuming). If you require more messages of the filtered type, increase the number of messages retrieved using 'number'.

Syntax: (critical|warning|debug|info)

Optional: True

number

Number of lines to retrieve from log.

Syntax: INT

Minimum: 1

Default: 20

show manufacture [verbose]

Show manufacturer information.

Parameters

verbose

Display more information (less concise, more detail).

Syntax: BOOLEAN

Default: False

Optional: True

show modem [verbose] [imei STRING] [name STRING]

Show modem status and statistics.

Parameters

imei

The IMEI of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

name

The configured name of the modem to execute this CLI command on.

Syntax: STRING

Optional: True

verbose

Display more information (less concise, more detail).

Syntax: BOOLEAN

Default: False

Optional: True

show nemo [name STRING]

Show NEMO status and statistics.

Parameters

name

The name of a specific NEMO instance.

show network [all|verbose] [interface STRING]

Show network interface status and statistics.

Parameters

all
Display all interfaces including disabled interfaces.
Syntax: BOOLEAN
Default: False
Optional: True

interface
Display more details and config data for a specific network interface.
Syntax: STRING
Optional: True

verbose
Display more information (less concise, more detail).
Syntax: BOOLEAN
Default: False
Optional: True

show openvpn

Show OpenVPN status and statistics.

openvpn client [*all*] [name *STRING*]

Show OpenVPN client status statistics.

Parameters

all
Display all clients including disabled clients.
Syntax: BOOLEAN
Default: False
Optional: True

name
Display more details and config data for a specific OpenVPN client.
Syntax: STRING
Optional: True

openvpn server [*all*] [name *STRING*]

Show OpenVPN server status and statistics.

Parameters

all
Display all servers including disabled servers.
Syntax: BOOLEAN
Default: False
Optional: True

name
Display more details and config data for a specific OpenVPN server.
Syntax: STRING
Optional: True

show route [ipv4|ipv6|verbose]

Show IP routing information.

Parameters

ipv4
Display IPv4 routes.
Syntax: BOOLEAN
Default: False
Optional: True

ipv6
Display IPv6 routes.
Syntax: BOOLEAN
Default: False
Optional: True

verbose
Display more information (less concise, more detail).
Syntax: BOOLEAN
Default: False
Optional: True

show scripts

Show scheduled system scripts

Parameters

None

show serial PORT

Show serial status and statistics.

Parameters

port

Display more details and config data for a specific serial port.

Syntax: STRING

Optional: True

show system [verbose]

Show system status and statistics.

Parameters

verbose

Display more information (disk usage, etc)

Syntax: BOOLEAN

Default: False

Optional: True

show usb

Show USB information.

Parameters

None

show version [verbose]

Show firmware version.

Parameters

verbose

Display more information (build date)

Syntax: BOOLEAN

Default: False

Optional: True

show vrrp [all|verbose] [name STRING]

Show VRRP status and statistics.

Parameters

all

Display all VRRP instances including disabled instances.

Syntax: {True|False}

Type: boolean

name

Display more details and configuration data for a specific VRRP instance.

Optional: True

Type: string

verbose

Display all VRRP status and statistics including disabled instances.

Syntax: {True|False}

Type: boolean

show web-filter

Show web filter status and statistics.

Parameters

None

ssh

Use SSH protocol to log into a remote server.

ssh [command STRING] host STRING [port INTEGER] user STRING

Parameters

command

The command that will be automatically executed once the SSH session to the remote host is established.

Optional: True

Type: string

host

The hostname or IP address of the remote host

Syntax: {*hostname*|*IPv4_address*|*IPv6_address*}

Type: string

port

The SSH port to use to connect to the remote host.

Default: 22

Maximum: 65535

Minimum: 1

Syntax: {*Integer*}

Type: integer

user

The username to use when connecting to the remote host.

Type: string

system

System commands.

system backup [passphrase STRING] type STRING PATH

Save the device's configuration to a file. Archives are full backups including generated SSH keys and dynamic DHCP lease information. Command backups are a list of CLI commands required to build the device's configuration.

Parameters

passphrase

Encrypt the archive with a passphrase.

Syntax: STRING

Optional: True

Depends on: **type** equals 'archive'

type

The type of backup file to create. Archives are full backups including generated SSH keys and dynamic DHCP lease information. CLI configuration backups are a list of CLI commands used to build the device's configuration.

Syntax: (cli-config|archive)

Default: archive

path

The file path to save the backup to.

Syntax: STRING

system disable-cryptography

Erase the device's configuration and reboot into a limited mode with no cryptography available. The device's shell will be accessible over Telnet (port 23) at IP address 192.168.210.1. To return the device to normal operation, perform the configuration erase procedure with the device's ERASE button twice consecutively.

Parameters

None

system duplicate-firmware

Duplicate the running firmware to the alternate partition so that the device will always boot the same firmware version.

Parameters

None

system factory-erase

Erase the device to restore to factory defaults. All configuration and automatically generated keys will be erased.

Parameters

None

system firmware update file STRING

Update the current firmware image. Upon reboot the new firmware will be run.

Parameters

file

Firmware filename and path.

Type: string

system restore [passphrase STRING] PATH

Restore the device's configuration from a backup archive or CLI commands file.

Parameters

path

The path to the backup file.

Syntax: STRING

passphrase

Decrypt the archive with a passphrase.

Syntax: STRING

Optional: True

system script stop SCRIPT

Stop an active running script. Scripts scheduled to run again will still run again (disable a script to prevent it from running again).

Parameters

script

Script to stop.

Syntax: STRING

system serial clear PORT

Clears the serial log.

Parameters

port

Serial port.

Type: string

system serial save PORT FILENAME

Saves the current serial log to a file.

Parameters

port

Serial port.

Type: string

filename

The filename to save the serial log. The file will be saved to the device's /etc/config/serial directory.

Type: string

system serial show PORT

Displays the serial log on the screen.

Parameters

port

Serial port.

Type: string

system serial start [size INTEGER] PORT

Start logging data on a serial port.

Parameters

size

Maximum size of serial log.

Default: 65536

Syntax: {*Integer*}

Type: integer

port

Serial port.

Type: string

system serial stop PORT

Start logging data on a serial port.

Parameters

port

Serial port.

Type: string

system support-report PATH

Save a support report to a file and include with support requests.

Parameters

path

The file path to save the support report to.

Syntax: STRING

traceroute

Print the route packets trace to network host.

traceroute [bypass|debug|dontfragment|icmp|ipv6|nomap] [first_ttl INTEGER] [gateway STRING] [interface STRING] [max_ttl INTEGER] [nqueries INTEGER] [packetlen INTEGER] [pausemsecs INTEGER] [port INTEGER] [src_addr STRING] [tos INTEGER] [waittime INTEGER] HOST

Parameters

bypass

Bypass the normal routing tables and send directly to a host on an attached network.

Syntax: BOOLEAN

Default: False

Optional: True

debug

Enable socket level debugging.

Syntax: BOOLEAN

Default: False

Optional: True

dontfragment

Do not fragment probe packets.

Syntax: BOOLEAN

Default: False

Optional: True

first_ttl

Specifies with what TTL to start.

Syntax: INT

Minimum: 1

Default: 1

gateway

Tells traceroute to add an IP source routing option to the outgoing packet that tells the network to route the packet through the specified gateway

Syntax: STRING

Optional: True

icmp

Use ICMP ECHO for probes.

Syntax: BOOLEAN

Default: False

Optional: True

interface

Specifies the interface through which traceroute should send packets. By default, the interface is selected according to the routing table.

Syntax: STRING

Optional: True

ipv6

If a hostname is defined as the value of the 'host' parameter, use the hosts IPV6 address.

Syntax: BOOLEAN

Default: False

Optional: True

max_ttl

Specifies the maximum number of hops (max time-to-live value) traceroute will probe.

Syntax: INT

Minimum: 1

Default: 30

nomap

Do not try to map IP addresses to host names when displaying them.

Syntax: BOOLEAN

Default: False

Optional: True

nqueries

Sets the number of probe packets per hop. A value of -1 indicated

Syntax: INT

Minimum: 1

Default: 3

packetlen

Total size of the probing packet. Default 60 bytes for IPv4 and 80 for Ipv6. A value of -1 specifies that the default value will be used.

Syntax: INT

Minimum: -1

Default: -1

pausemsecs

Minimal time interval between probes

Syntax: INT

Minimum: 0

Default: 0

port

Specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). A value of -1 specifies that no specific port will be used.

Syntax: INT

Minimum: -1

Default: -1

src_addr

Chooses an alternative source address. Note that you must select the address of one of the interfaces. By default, the address of the outgoing interface is used.

Syntax: STRING

Optional: True

tos

For IPv4, set the Type of Service (ToS) and Precedence value. Useful values are 16 (low delay) and 8 (high throughput). Note that in order to use some TOS precedence values, you have to be super user. For IPv6, set the Traffic Control value. A value of -1 specifies that no value will be used.

Syntax: INT

Minimum: -1

Default: -1

waittime

Determines how long to wait for a response to a probe.

Syntax: INT

Minimum: 1

Default: 5

host

The host that we wish to trace the route packets for.

Syntax: STRING

config service anywhereusb enable

config service anywhereusb enable true|false

Allow remote access to USB devices connected to this server. The default **TCP Port** value is 18574.

config service anywhereusb port

config service anywhereusb port {1-65535}

Specify the port number that is used to access the Hub. The default value is 18574. If you change the port number you must also change the corresponding port number on your computer.

config service anywhereusb groups

Assign a name to each group and specify the ports in each group. When a client connects to a group in the **AnywhereUSB Manager**, the user has access to all of the ports in the group.

You can change the name for a group in the **Group Description** field. By default, a group is named "Group" appended by a consecutive number, such as Group 1, Group 2, and so on. This name displays in the **Group Name** field in the Group Status pane.

For each group, you can specify ports.

Note Each port should be assigned to only one group.

You can also do this in the web UI. See Create groups and assign ports to the group.

Syntax

```
config service anywhereusb groups [option]
```

Options

group(01-24) description "*string*"

Enter a name for the group. Replace *string* with the group name. You must have double quotes around the name.

group(01-24) ports (0-23) (1-24)

Specify group number to change and a single port or a range of ports to assign to this group.

Note Ports can only be assigned to one group at a time. If a port is assigned to a new group, it is removed from the current group.

Examples

Specify a group name for group 2

```
config service anywhereusb groups group02 description "Group 2 name"
```

Replace an existing port assignment

Replace the group 1 port at index 0 with port 1.

```
config service anywhereusb groups group01 ports 0 1
```

View current port settings

In this example, there are three assigned ports: port 1 (occupying index position 0), port 2 (index position 1) and port 3 (index position 2).

```
config show service anywhereusb groups group01 ports
0  1
1  2
2  3
```

Delete a port from a group

In the previous example, there are three assigned ports in group 1: port 1 (occupying index position 0), port 2 (index position 1) and port 3 (index position 2). This example shows how to delete ports 2 and 3, leaving only port 1 in this group. Ports are deleted by index number, not port number.

```
config del service anywhereusb groups group01 ports 1
config del service anywhereusb groups group01 ports 2
```

Add a port to the first available index number

Add port 1 to the first available index number.

```
config add service anywhereusb groups group01 ports end 1
```

Reassign ports based on the port's index number

In this example, one port is defined in the group: port 2 (occupying index position 0):

```
config show service anywhereusb groups group01 ports
0  2
```

You can change this port designation to "1". The syntax here changes the value of the index 0 item to port 1.

```
config service anywhereusb groups group01 ports 0 1
```

config service anywhereusb clients

Add a client ID to the client list. When a computer searches for Hubs, any computer with a client ID on the client list can connect to the Hub. You can also add client IDs in the web UI. See Manually add a client ID.

Syntax

```
config service anywhereusb clients [option]
```

Options

0-255

Specify the client index.

[id "*string*"]

Specify the client ID for the computer.

[description "*string*]"

Specify a descriptive name for the computer.

groups (0-23) (group01-24)

Specify the groups this client ID can access.

Examples

You must be in configuration mode to use these commands.

Show a list of clients

This command shows the client description, the groups assigned to the client, and the client ID for each client.

```
> config
(config) > show service anywhereusb clients
0
   description Client description
    groups
          0 group01
          1 group02
    id Client_ID
......
```

Add a new client

A new elements is added before the given index. You can add "end" with the index to add the new client to the end of the array. Specifying a client ID is required. Other fields are optional.

```
> config
(config)> add service anywhereusb clients (0-254|end)
(config service anywhereusb clients 0)> id "Client_ID"
(config service anywhereusb clients 0)> save
```

Replace a group

This example replaces the group at index 0 with group 2. The client must have at least one group already assigned.

```
config service anywhereusb clients 0 groups 0 group02
```

Delete a client

You must specify the index of the client (0-254) to delete it.

```
> config
(config)> del service anywhereusb clients (6)
(config)> save
```

USEALLHUBADDRS

Enable or disable the **AnywhereUSB Manager** from connecting to extra IP addresses.

The AnywhereUSB Hub may have default IP addresses that are reported by mDNS to the **AnywhereUSB Manager**, but in many network environments, the **Manager** cannot connect to them. As part of normal operation, the **Manager** tries to sequentially connect to all of the Hub IP addresses, so if it starts trying these extra default IP addresses, it may take extra time (minutes) for the **Manager** to connect or reconnect.

By default, this option is deselected and the **Manager** does not attempt to connect to these addresses.

Note This can also be done in the **Preferences** dialog. See Use all Hub addresses.

USEALLHUBADDRS,[on|off]

Parameters

- **off**: Disable the **Use All Hub Addresses** option. The **AnywhereUSB Manager** will not attempt to connect to the extra IP addresses. This is the default.
- **on**: Enable the **Use All Hub Addresses** option. The **AnywhereUSB Manager** will attempt to connect to the extra IP addresses.

Configure the AnywhereUSB Manager from the command line

You can configure and control the **AnywhereUSB Manager** from the command line.

Prerequisites

- **Service**: If you run the **AnywhereUSB Manager** as a service, you need to be an Administrator.
- **Stand-alone**: If you run the **AnywhereUSB Manager** as a stand-alone, you need to be the same user that was logged in during the installation process or an Administrator. If you are not an Administrator, and you try to configure the **AnywhereUSB Manager** from a command line with the "Run as Administrator" option selected, the commands will fail.

autoconnect clear all

Purpose

Disables the auto connect feature for all Hubs, groups, and devices. When complete no asterisks or plus signs display next to Hub, group, or device names.

Syntax

```
>awusbmanager autoconnect clear all
```

Examples

Run the list command to verify the current state of the auto-connect feature for the Hubs, groups, and devices. In this example, Group 1 has auto connect enabled, and the device in Group 1 has inherited the auto connect feature.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2)
*  Group 1 (AW02-000001.1) (In-use by you)
+       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `autoconnect clear all` command.

```
>awusbmanager autoconnect clear all
```

Run the list command again to verify that the auto connect feature has been disabled. No asterisks or plus signs should display.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2)
   Group 1 (AW02-000001.1) (In-use by you)
        U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

autoconnect clear group

Purpose

Disable the auto connect feature for a specified group.

When you disable auto connect for a group, an asterisk no longer displays next to the group name. In addition, any devices in the group no longer inherit the auto connect feature, and the plus sign no longer displays next to the device names.

Note For more information about auto connect, see Configure auto connect.

Syntax

```
>awusbmanager autoconnect clear group,<address>
```

The [*address*] is the address of the group for which you want to disable the auto connect feature.

Examples

Run the list command to verify the current state of the auto-connect feature for a group and to determine the address for a group. In this example, Group 1 has the auto connect feature enabled, so an asterisk displays next to the group name.

The [*address*] for a group is the name of the Hub appended by the number of the group. In this example, the auto connect feature will be disabled for Group 1, so the group name is highlighted below.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2)
*  Group 1 (AW02-000001.1) (In-use by you)
+       U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `autoconnect clear group` command.

```
>awusbmanager autoconnect clear group,AW02-000001.1
```

Run the list command again to verify that the auto connect feature has been disabled. In this example, the auto connect feature has been disabled for Group 1, so an asterisk no longer displays next to the group name. In addition, the plus sign no longer displays next to the devices in Group 1.

Note If you were connected to the group and the devices in the group, you will still be connected. If you want do disconnect from them, you can use the `disconnect group` command.

```
AnywhereUSB Manager, below are the available devices:
```

```
AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2)
   Group 1 (AW02-000001.1) (In-use by you)
        U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

autoconnect group

Purpose

Enable the auto connect feature for a specified group. This feature ensures that when you start the **AnywhereUSB Manager** as a stand-alone or when it starts at Windows startup if installed as a service, you are automatically connected to all of the groups to which you are allowed access that have auto connect enabled.

When you enable auto connect for a group, an asterisk displays next to the group name. In addition, any devices in the group inherit the auto connect feature, and will also be automatically connected. A plus sign displays next to the devices when the auto connect feature is inherited.

You can disable the auto connect feature for the group if needed.

Note For more information about auto connect, see Configure auto connect.

Syntax

```
>awusbmanager autoconnect group,<address>
```

The [*address*] is the address of the group for which you want to enable the auto connect feature.

Examples

Run the list command to verify the current state of the auto-connect feature for a group and to determine the address for a group. In this example, Group 2 has the auto connect feature enabled, so an asterisk displays next to the group name. The auto connect feature is not enabled for Group 1, so an asterisk does not display.

The [*address*] for a group is the name of the Hub appended by the number of the group. In this example, the auto connect feature will be enabled for Group 1, so the group name is highlighted below.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
*  Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the the `autoconnect group` command.

```
>awusbmanager autoconnect group,AW02-000001.1
```

Run the list command again to verify that the auto connect feature has been enabled. An asterisk displays next to the group name. A plus sign displays next to the names of the devices in the group to show that the auto connect feature is inherited from the group.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
*  Group 2 (AW02-000001.2) (In-use by you)
*  Group 1 (AW02-000001.1) (In-use by you)
+      U3 Cruzer Micro (AW02-000001.1101(  (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

autofind

Purpose

Enables and disables the autofind feature. When enabled, all Hubs connected to the network when **AnywhereUSB Manager** launches are automatically found. This command works as a toggle, or you can can specify "on" or "off." Before you used the command, you should verify the status of the autofind feature.

The status of the autofind feature is displayed when you run the list command.

Note For information about this feature in the **AnywhereUSB Manager**, see Autofind Hubs in the AnywhereUSB Manager.

Syntax

```
>awusbmanager autofind[,on|,off]
```

- **on**: Enables the autofind feature. When enabled, all Hubs connected to the network when **AnywhereUSB Manager** launches are automatically found. This option is not required.
- **off**: Disables the autofind feature. When disabled, Hubs are not automatically found when **AnywhereUSB Manager** launches. In this case, you must manually add the Hubs to which you want to connect to the known Hubs list. This option is not required.

Examples

Run the list command to verify the status of the autofind feature. In this example, the autofind feature is enabled.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the autofind command to disable the feature. You can specify the "off" option, but it is not required.

```
>awusbmanager autofind,off
```

Run the list command again.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: disabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

You can run the autofind command again to enable the feature. You can specify the "on" option, but it is not required.

```
>awusbmanager autofind
```

Run the list command again to verify.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

connect device

Purpose

Connect to a USB device in a group to which you have access. You cannot connect to a device in a group that is already in use.

You must be connected to the group before you can connect to a device in that group. If the device requires a password, you must enter the password when using the `connect device` command.

Syntax

```
>awusbmanager connect device,<address>[,password]
```

The <*address*> is the address of the device to which you want to connect.

The <*password*> is the password for the device. The password is optional.

Examples

If you have connected to a group, and then disconnect from a device in that group, you no longer have access to the device. You can reconnect to that device.

Run the list command to make sure you are connected to the group that the device you want to connect to is in. In this example, the device is in Group 1, so you should be connected to Group 1.

You will need the address for device to which you want to connect.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `connect device` command. If required to access the device, include the device password.

```
>awusbmanager connect device,AW02-000001.1101
```

Run the list command again to verify that the device is connected.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
```

```
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

connect group

Purpose

You can connect to a group so that you have access to the ports in the group. Once you have connected to a group, no one else can connect to that group. You cannot connect to a group that is already is use.

When you connect to a group, you are automatically connected to all of the ports in the group to which you are allowed access.

Syntax

```
>awusbmanager connect group,<address>
```

The [*address*] is the address of the group to which you want to connect.

Examples

Run the list command to determine the address for the group to which you want to connect. In this example, you will connect to Group 1.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `connect group` command.

```
>awusbmanager connect group,AW02-000001.1
```

Run the list command again to verify that you are connected to the group and to all of the ports in the group to which you are allowed access.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

device info

Purpose

Displays information about a device. For more information, see AnywhereUSB Manager Device Status pane.

Syntax

```
>awusbmanager device info,<address>
```

The [*address*] is the address of the device for which you want to display information. The address is required.

Examples

Run the list command to determine the device's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
*  Group 2 (AW02-000001.2) (In-use by you)
*  Group 1 (AW02-000001.1) (In-use by you)
+      U3 Cruzer Micro "USB stick 1" (AW02-000001.1101)  (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `device info` command.

```
>awusbmanager device info,AW02-000001.1101
```

Information about the device displays.

```
ADDRESS: AW02-000001.1101
LOCALNAME: USB stick 1
VENDOR: SanDisk
VENDOR ID: 0x0781
PRODUCT: U3 Cruzer Micro
PRODUCT ID: 0x5406
SERIAL: 0770000F0000000C
AUTOCONNECT: inherited
IN USE BY: YOU
```

device name

Purpose

Change or assign the local name of a device.

Syntax

```
>awusbmanager device name,<address>,<new name>
```

The <*device name*> is the device's address.

The <*new name*> is the new local name for the device.

Examples

Run the list command to determine the device's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `device name` command.

```
>awusbmanager device name,AW02-000001.1101,USB Stick
```

Run the list command again to verify the name change.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro "USB Stick" (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

disconnect device

Purpose

Disconnect from a USB device to which you no longer need access. You will remain connected to the group that the device is in. Other users cannot connect the USB device, since you still own the group that the USB device is in.

Note If you have auto connect enabled for the group, you can disconnect from a USB device in the group, but note that for a stand-alone, the device will be reconnected the next time you stop and then start the **AnywhereUSB Manager**. If you are running the **AnywhereUSB Manager** as a service, the device remains disconnected until you reboot the PC or restart the service.

Syntax

```
>awusbmanager disconnect device,<address>
```

The <*address*> is the address of the device from which you want to disconnect.

Examples

Run the list command to view the address for device from which you want to disconnect.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `disconnect device` command.

```
>awusbmanager disconnect device,AW02-000001.1101
```

Run the list command again to verify that the device is disconnected.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

disconnect group

Purpose

You can disconnect from a group that has ports you no longer need access to. You are disconnected from all USB devices and ports in that group. Any other user can then connect to that group.

Note If the group has auto connect enabled, and you want to disconnect from the group, note that when you disconnect from the group you will be automatically reconnected. Before you disconnect, make sure that auto connect is disabled for the group.

Syntax

```
>awusbmanager disconnect group, [address]
```

The [*address*] is the address of the group from which you want to disconnect.

Examples

Run the list command to determine the address for the group to which you want to connect.

Make sure that auto connect is disabled for the group. When it is disabled, an asterisk does not display next to the group name. If you need to disable auto connect for the group, see autoconnect clear group.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101) (In-use by you)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `disconnect group` command.

```
>awusbmanager disconnect group,AW02-000001.1
```

Run the list command again to verify that the group is disconnected.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
```

```
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

exit

Purpose

Shuts down the service. If the **AnywhereUS Manager** is open, it is shut down as well.

Syntax

```
>awusbmanager exit
```

group info

Purpose

Displays information about a group. For more information, see AnywhereUSB Manager Group Status pane.

Syntax

```
>awusbmanager group info,[address]
```

The [*address*] is the address of the group for which you want to display information. The address is required.

Examples

Run the list command to determine the group's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "HUB-000001" (AW02-000001.local.:18574)
*  Group 2 "Admin group" (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `group info` command.

```
>awusbmanager group info,AW02-000001.2
```

Information about the group displays.

```
ADDRESS: AW02-000001.2
LOCALNAME: Admin group
GROUP: 2
NAME: Group 2
PORTS: 2
AUTOCONNECT: enabled
IN USE BY: YOU
```

group name

Purpose

Change or assign the local name of the group.

Syntax

```
>awusbmanager group name,<address,<new name>
```

The <*group name*> is the group's address.

The <*new name*> is the new local name for the group.

Examples

Run the list command to determine the group's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `group name` command.

```
>awusbmanager group name,AW02-000001.2,New Group
```

Run the list command again to verify the name change.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 "New Group" (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

hidden hub add

Purpose

Hide a Hub by adding it to the hidden Hubs list.

Note For information on hiding Hubs in the AnywhereUSB Manager, see Hide an individual Hub and Hide all unauthorized Hubs.

Syntax

```
>awusbmanager hidden hub add,<address>[:port]
```

The [*address*] is the address of the Hub that you want to hide

The <*port*> is the TCP port number for the Hub you want to hide. This is required if the TCP port number is not the default (18574).

Examples

Run the `hidden hub add` command to add a Hub to the hidden Hub list.

- Use the default port of 18574:

```
>awusbmanager hidden hub add,10.10.10.34
```

- Change the TCP port number:

```
>awusbmanager hidden hub add,10.10.10.56:5600
```

You can then run the `hidden hub list` command to verify that the Hubs were added to the list of hidden Hubs.

```
10.10.10.34:18574
10.10.10.56:5600
```

hidden hub list

Purpose

Displays a list of Hubs that have been added to the hidden Hubs list.

- You can choose to hide Hubs that currently display in the **AnywhereUSB Manager**, such as an unauthorized Hub (which displays with a red X next to the Hub name), or a Hub which users shouldn't access.
- You can also choose to hide Hubs that don't currently display in the **AnywhereUSB Manager**, but the client ID may have access in the future, such as a Hub on another network.

Note For information on hiding Hubs in the **AnywhereUSB Manager**, see Hide an individual Hub and Hide all unauthorized Hubs.

Syntax

```
>awusbmanager hidden hub list
```

Examples

Run the `hidden hub list` command.

```
>awusbmanager hidden hub list
```

A list of hidden Hubs is returned.

```
10.10.10.50:18574
10.10.10.21:18574
```

hidden hub remove

Purpose

Remove a Hub from the hidden Hubs list.

Syntax

```
>awusbmanager hidden hub remove,<address>[:port]
```

The <*address*> is the address of the hub that you want to remove from the hidden Hub list. This is required.

The <*port*> is the TCP port number for the Hub you want to remove. This is required if the TCP port number is not the default (18574).

Examples

Run the `hidden hub list` command to verify the address and port number of the Hub that you want to remove.

```
10.10.10.21:18574
10.10.10.34:18574
10.10.10.56:5600
```

Run the `hidden hub remove` command.

- If the TCP port number is the default, entering the port number in the command is optional.

```
>awusbmanager hidden hub remove,10.10.10.34
```

- If the TCP port number is not the default, entering the port number in the command is required.

```
>awusbmanager hidden hub remove,10.10.10.56:5600
```

Run the `hidden hub list` command again to verify that the specified Hubs have been removed.

```
10.10.10.21:18574
```

hidden hub remove all

Purpose

Remove all the Hubs in the hidden Hubs list.

Syntax

```
>awusbmanager hidden hub remove all
```

Examples

Run the `hidden hub list` command to view the list of hidden Hubs.

```
10.10.10.12:18574
10.10.10.14:18574
10.10.10.15:5600
```

Run the `hidden hub remove all` command.

```
>awusbmanager hidden hub remove all
```

Run the `hidden hub list` command again to verify that the Hubs have been removed.

help

Purpose

Displays a list of the CLI commands for the **AnywhereUSB Manager**.

Syntax

```
>awusbmanager help
```

hub info

Purpose

Displays information about the Hubs. For more information, see AnywhereUSB Manager Hub Status pane.

Syntax

```
>awusbmanager hub info,<hub name>
```

The <*hub name*> is the address of the Hub for which you want to display information. The address is required.

Examples

Run the list command to determine Hub's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "HUB-000001" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `hub info` command.

```
>awusbmanager hub info,AW02-000001
```

Information about the Hub displays.

```
NAME: AW02-000001
LOCALNAME: HUB-000001
MODEL: AnywhereUSB 2 Plus
VERSION: 3.0.0.54 awusb dby-3.0.0.54 01/03/2019 16:44:25 CST 20190103224522
STATE: Active (secure)
ADDRESS: AW02-000001.local. (SSL Subject:/C=US/ST=Minnesota/O=Digi International
Inc/CN=unknown ,Issuer:/C=US/ST=Minnesota/O=Digi International Inc/CN=unknown)
(10.10.74.xxx)
PORT: 18574
CONNECTED FOR: 22115 sec
CONNECTION ID: 1
INTERFACE: eth0
SERIAL NUMBER: AW02-000001
AUTOCONNECT: disabled
```

hub name

Purpose

Change or assign the local name of the Hub.

Syntax

```
>awusbmanager hub name,<address[:port]>,<new name>
```

The <*address*> is the Hub's address.

The <*port*> is the TCP port number for the Hub you want to rename. This is required if the TCP port number is not the default (18574).

The <*new name*> is the new local name for the Hub.

Examples

Run the list command to determine the Hub's address.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

Run the `hub name` command.

```
>awusbmanager hub name,AW02-000001,Hub 1
```

Run the list command again to verify the local name.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

known hub add

Purpose

Add a Hub to the known Hubs list. The Hubs in this list can be on the same network as your computer, or on a different network. If you add Hubs to the known Hubs list that are on the same network as our computer AND the autofind feature is enabled, duplicate entries display in the Hubs list.

> **Note** For information about using this feature in the **AnywhereUSB Manager**, see Manage the list of known Hubs.

Syntax

```
>awusbmanager known hub add,<address>[:port]
```

The <*address*> is the address of the Hub or a Hub hostname that can be resolved by your network nameservers. This is required.

The <*port*> is the TCP port number, which is 18574 by default. You can change the TCP port number if needed.

Examples

Add a known Hub

Run the `known hub add` command to add a Hub to the known Hub list.

- Use and address and the default port of 18574:

```
>awusbmanager known hub add,10.10.56.12
```

- Use a hostname and change the TCP port number:

```
>awusbmanager known hub add,awusb1.work.com:9999
```

- Change the TCP port number:

```
>awusbmanager known hub add,10.10.56.14:5600
```

You can then run the `known hub list` command to verify that the Hub was added to the list.

```
10.10.10.56:18574
awusb1.work.com:9999
10.10.56.14:5600
```

known hub list

Purpose

Displays a list of Hubs that have been added to the known Hubs list.

Note For more information about known Hubs, see Manage the list of known Hubs.

Syntax

```
>awusbmanager known hub list
```

Examples

Run the `known hub list` command.

```
>awusbmanager known hub list
```

A list of known Hubs is returned.

```
10.10.10.50:18574
10.10.10.12:18574
```

known hub remove

Purpose

Remove a Hub from the known Hubs list.

Note For information about using this feature in the **AnywhereUSB Manager**, see Manage the list of known Hubs.

Syntax

```
>awusbmanager known hub remove,<address>[:port]
```

The <*address*> is the address of the hub that you want to remove from the known Hub list. This is required.

The <*port*> is the TCP port number for the Hub you want to remove. This is required if the TCP port number is not the default (18574).

Examples

Run the `known hub list` command to verify the address and port number of the Hub that you want to remove.

```
10.10.01.12:18574
10.10.01.14:18574
10.10.01.15:5600
```

Run the `known hub remove` command.

- If the TCP port number is the default, entering the port number in the command is optional.

```
>awusbmanager known hub remove,10.10.01.14
```

- If the TCP port number is not the default, entering the port number in the command is required.

```
>awusbmanager known hub remove,10.10.01.15:5600
```

Run the `known hub list` command again to verify that the Hubs have been removed.

```
10.10.01.12:18574
```

known hub remove all

Purpose

Remove all the Hubs in the known Hubs list.

Syntax

```
>awusbmanager known hub remove all
```

Examples

Run the `known hub list` command to view the list of known Hubs.

```
10.10.01.12:18574
10.10.01.14:18574
10.10.01.15:5600
```

Run the `known hub remove all` command.

```
>awusbmanager known hub remove all
```

Run the `known hub list` command again to verify that the Hubs have been removed.

list

Purpose

Displays a list of Hubs, groups, and devices on the network as well as any Hubs the **AnywhereUSB Manager** knows about.

Note This information is similar to what displays in the **AnywhereUSB Manager**. See AnywhereUSB Manager window.

If a group has auto connect enabled, an asterisk displays next to the group name.

Additional information about features displays at the bottom of the list:

- Status of the auto-find feature: enabled or disabled.
- Status of the **auto connect all** feature: enabled or disabled.
- Specifies whether the **AnywhereUSB Manager** is running as a service.

Syntax

```
>awusbmanager list
```

Examples

This example shows one Hub: AW02-000001. On the Hub, Group 1 has the auto connect feature enabled, as specified by the asterisk next to the group name.

```
AnywhereUSB Manager, below are the available devices:

AW02-000001 "Hub 1" (AW02-000001.local.:18574)
   Group 2 (AW02-000001.2) (In-use by you)
   Group 1 (AW02-000001.1) (In-use by you)
       U3 Cruzer Micro (AW02-000001.1101)

* means Autoconnect enabled, + means Autoconnect inherited
Auto-Find: enabled
Autoconnect All: disabled
AnywhereUSB Manager not running as a service
```

list full

Purpose

Displays a list of all Hubs, groups, and devices on the network and includes all information about each Hub, group, or device. This command displays the same information retrieved by running these commands: list, hub info, group info, and device info.

If a group has auto connect enabled, an asterisk displays next to the group name.

Additional information about features displays at the bottom of the list:

- Status of the auto-find feature: enabled or disabled.
- Status of the **auto connect all** feature: enabled or disabled.
- Specifies whether the **AnywhereUSB Manager** is running as a service.

Syntax

```
>awusbmanager list full
```

Examples

Run the `list full` command.

```
>awusbmanager list full
```

The example below shows the Hub on the network, and the groups and devices on that Hub. Information about the Hub, group, and device is also returned.

```
AnywhereUSB Manager, below are the available devices:

AW08-D00001 (10.10.12.12:18574)
  NAME: AW08-D00001
  MODEL: AnywhereUSB 8 Plus
  VERSION: 3.0.1.2 awusb
  STATE: Active (secure)
  ADDRESS: 10.10.12.12
  PORT: 18574
  CONNECTED FOR: 14 sec
  CONNECTION ID: 3
  INTERFACE: eth0
  SERIAL NUMBER: AW08-D00001
  AUTOCONNECT: disabled

   Group  2 (AW08-D00001.2)
     ADDRESS: AW08-D00001.2
     GROUP: 2
     NAME: Group  2
     PORTS: 5 6 7 8
     AUTOCONNECT: disabled
     IN USE BY: NO ONE

      Cruzer (AW08-D00001.1906)
        ADDRESS: AW08-D00001.1906
        VENDOR: SanDisk
        VENDOR ID: 0x0781
        PRODUCT: Cruzer
        PRODUCT ID: 0x5530
        SERIAL: 20040000920A1C707B00
        AUTOCONNECT: disabled
        IN USE BY: NO ONE


*  Group  1 (AW08-D00001.1) (In-use by you)
     ADDRESS: AW08-D00001.1
     GROUP: 1
     NAME: Group  1
     PORTS: 1 2 3 4
     AUTOCONNECT: enabled
     IN USE BY: YOU

+     USB DISK 3.0 (AW08-D00001.1803) (In-use by you)
        ADDRESS: AW08-D00010.1803
        VENDOR:
        VENDOR ID: 0x13fe
        PRODUCT: USB DISK 3.0
        PRODUCT ID: 0x6300
        SERIAL: 070A00376967E000
        AUTOCONNECT: inherited
        IN USE BY: YOU

* means Autoconnect enabled, + means Autoconnect inherited
Autofind: disabled
Autoconnect All: disabled
AnywhereUSB Manager is running as a service
```

Security

Security-related features in AnywhereUSB include:

- Unique password for each Hub. See Change the Hub password.
- Configurable network service port numbers.
- Secure access and authentication to the web UI and CLI.
- One password, one permission level.
- Selectively enable and disable network services such as mDNS, HTTP/HTTPS, and SSH.
- Encrypted access to AnywhereUSB® Plus traffic: Access to the USB-over-IP traffic is encrypted and authenticated by default. This cannot be disabled.

Client ID

The client ID is a unique identifier that you assign when you initially install the **Anywhere USB Manager**. When you launch the **Manager** for the first time and log in, the **Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your account with the Hub.

- **Stand-alone**: If you installed the **Manager** as a stand-alone, the client ID and the certificate identify the user's login credentials on the computer.
- **Service**: If you installed the **Manager** as a service, the client ID and the certificate identify the computer.

When the client ID and certificate have been created, the computer is able to connect to the Hubs that recognize that client ID. Any other computer with the same client ID will be rejected.

Note In some cases, multiple computers may inadvertently be used by multiple users that have the same client ID. To fix this issue, see AnywhereUSB Manager client ID is not unique.

Client ID length

The number of characters allowed in the **Client ID** field is variable and is dependent on UTF-8 encoding of the characters. Note that some characters are multi-byte characters, which reduces the number of characters that are allowed in the field. Currently, the **Client ID** field is a maximum of 63 bytes encoded in UTF-8.

Assign a client ID to a user account

A client ID is assigned to user credentials the first time a user logs into a computer and launches the **AnywhereUSB Manager**.

WARNING! Digi recommends that you use a private network to connect the computer to the Hub. This ensures that only clients IDs with known user credentials can connect to the Hub. The first time that a client ID on a computer connects to the Hub, the unique credentials for this known user are stored in your Hub. If do not use a private network, an unknown computer with the same client ID may happen to connect to the Hub before the known computer connects. In this case, the known computer will not be able to connect and authenticate.

Step 1: Create a client ID during initial launch of the AnywhereUSB Manager

The **AnywhereUSB Manager** can be initially opened by a user in one of the following ways:

- **Installation**: When the AnywhereUSB Hub software is installed, the **Launch AnywhereUSB Manager** option is selected by default. When the installation completes, the client ID confirmation dialog appears. The user enters a client ID, and then the **AnywhereUSB Manager** is automatically launched.

  Note If the user deselects the **Launch AnywhereUSB Manager** option during installation, the **AnywhereUSB Manager** does not automatically open after the installation process completes. In this case, the client ID dialog does not display.

- **New user logs in**: After the AnywhereUSB Hub software is installed, any user can log into that computer and open the **AnywhereUSB Manager**. The first time a new user opens the **AnywhereUSB Manager**, the client ID dialog appears. The user must enter a client ID before the **AnywhereUSB Manager** will open.

After the initial launch of the **AnywhereUSB Manager**, the next time the user logs in, the computer is able to connect to the Hubs that recognize that client ID.

Step 2: Manually add a client ID to the client ID list in the Hub

You can manually add a client ID to the client list before a new user launches the **AnywhereUSB Manager** for the first time. In this situation, the certificate is unavailable until the first time a computer with the new client ID connects to the Hub. The new client ID is associated with the credentials for the user currently logged on to the computer.

When the computer connects to the Hub for the first time, the identity certificates are exchanged between the computer and the Hub. After the initial connection, only that computer with the client ID and unique identity certificate is able to connect to the Hub.

Troubleshooting

The following information provides troubleshooting steps for the most common issues. To find information on other issues, visit our Knowledge Base at knowledge.digi.com.

If you need to gather log files and other information, you can use the Create Support File feature.

AnywhereUSB Manager client ID is not unique

During the initial installation of the **Anywhere USB Manager**, you are required to assign a unique client ID. When you launch the **Manager** for the first time and log in, the **Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your account with the Hub.

- **Stand-alone**: If you installed the **Manager** as a stand-alone, the client ID and the certificate identify the user's login credentials on the computer.
- **Service**: If you installed the **Manager** as a service, the client ID and the certificate identify the computer.

Note See Client ID for more information about how the client ID is used by your computer and the Hub to create a connection.

In some cases, multiple computers may inadvertently be used by multiple users that have the same client ID. When this occurs, and computers with the same client ID attempt to connect with the same Hub, the first computer to associate itself with the Hub will be able to connect to the Hub. Subsequent computers with the same client ID will not be able to connect to that Hub.

You can fix this issue by changing the client ID of your computer to a unique client ID. See Change the client ID.

No remote Hubs found

When the host computer is unable to discover any AnywhereUSB devices on the network, no Hubs are displayed in the **AnywhereUSB Manager**.

Firewall software blocks the port used for Hub discovery

When firewall software blocks the port used for Hub discovery, try the following:

- For firewall software, either disable it or add an exception for the port (UDP port 5353).
- Check for a link light on the Ethernet port. If the link light is not lit, connect all of the Hubs to switches using network cables.
- Verify that the **Autofind Hubs** option is selected in the **Preferences** dialog in the **AnywhereUSB Manager**. Start the Manager and choose **File > Preferences** to open the dialog.
- Connect the Hub directly to the host computer.
- Some anti-virus software might block the connection. You can either temporarily disable it or add an exception for the **AnywhereUSB Manager** executable.
- If the Hub is across a switch or router that does not forward Bonjour traffic, the **AnywhereUSB Manager** will not be able to discover the Hub. In this case, add the Hub to the known Hubs list. See Manage the list of known Hubs.
- The firewall or router may block access to the AnywhereUSB port, which by default is TCP port 18574. If the Hub can be discovered but the connection fails (the state of the connection is "Unable to connect"), you may need to reopen the AnywhereUSB port.

Hide a group in the AnywhereUSB Manager

Any group that has ports assigned to it displays in the **AnywhereUSB Manager**, even if no USB devices are connected to a port. If you don't want groups with unused ports to display in the **AnywhereUSB**

Manager, you can reassign all of the ports in a group to a different group. Once the group does not have any ports assigned to it, that group will not display.

1. Open the web UI from your selected Hub.
2. Click **AnywhereUSB** from the **Configuration** section. The **AnywhereUSB Configuration** page appears.
3. Locate the group that has the unused ports.
4. Reassign each port in the group to a different group, or to the **Unassigned** row.
5. When done, click **Apply** to save the changes.
6. Return to the **AnywhereUSB Manager**. The group no longer appears.

Services turned off and locked out of the Hub

If you turn off services, be aware that if you turn off all of the services and the web UI, you will be locked out of the Hub and unable to access it.

If this happens, follow the process below to reconnect to the Hub.

1. Remove the Hub from the deployed network.
2. Press the RESET button on the Hub to restore the factory defaults.
3. Step 5: Connect to the device using an Ethernet LAN connection.
4. Step 6: Verify initial connection.
5. Configure the Hub settings.
6. Reconnect the Hubs to the existing **AnywhereUSB Managers**.

Microsoft Windows restrictions

Microsoft Remote Desktop

Some devices (such as a web camera), and some input devices (such as a USB keyboard or a mouse), are blocked and may not display when Microsoft Remote Desktop is connected to a laptop or a virtual machine.

For example, laptop A is connected to an AnywhereUSB Hub on the network, and a web camera is connected to a port on the Hub. Laptop A is able to see the video feed from the camera.

A user on laptop B can use Microsoft Remote Desktop to gain access to laptop A. In this situation, the video feed for both laptop A and laptop B is restricted by Windows and neither user can view the video feed from the web camera.

Hubs and virtual machines

Hubs may not function properly when attached to a Guest OS on a virtual machine.

To resolve this issue, ensure that the extensions for the virtual machine have been installed on the Guest OS.

Allow remote access to USB devices

You can configure the Hub to allow remote access to USB devices connected to this server. You must specify the port number that is used to access the Hub.

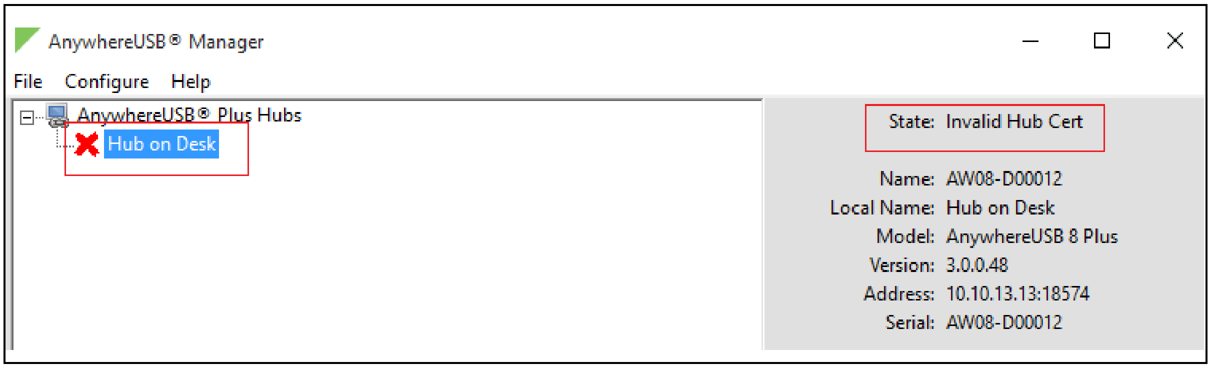
1. Open the web UI.
2. Select **System** > **Configuration** > **AnywhereUSB Configuration**. The **AnywhereUSB Configuration** page appears.
3. Select **Enable**.
4. Enter the port number in the **Port** field. The default **TCP Port** value is 18574. If you change the port number on this page, you must also change the corresponding port number on your computer.
5. Click **Apply** to apply and save the changes.

Invalid Hub certificate

In some situations, the Hub certificate may become invalid. If this occurs, you can remove the Hub from the Manage Hub Credentials list. The **AnywhereUSB Manager** forgets the Hub certificate and, if the **Auto-register Hub Cert** option is selected in the **Preferences** dialog, gets a new one on the next connection attempt.

If the **Auto-register Hub Cert** option in the **Preferences** dialog is not selected, you can manually add the Hub to the Manage Hub Credentials list. After it is added, the **AnywhereUSB Manager** gets a new certificate for the Hub on the next connection attempt.

Note The Hub must be on a secure network before you manually add the Hub to the Manage Hub Credentials list or if you remove the certificate and a new one is automatically assigned over the network.



State: Invalid Hub Cert

Name: AW08-D00012
Local Name: Hub on Desk
Model: AnywhereUSB 8 Plus
Version: 3.0.0.48
Address: 10.10.13.13:18574
Serial: AW08-D00012

Invalid client certificate

The client ID is a unique identifier assigned to a user account the first time a user logs in to a computer and opens the **AnywhereUSB Manager**. The client ID is associated with the login credentials for the user currently logged on to the computer.

During initial log in process, the **AnywhereUSB Manager** creates a secure identity certificate that is associated with the client ID. This certificate is used to validate your user account with the Hub. The certificate associated with the user account client ID must match the certificate for this client ID on the Hub to allow a connection.

In some situations, a mismatch occurs between the certificate associated with the client ID and certificate for the client ID on the Hub. When this happens, the message "Invalid client cert" displays as the **State** in the **AnywhereUSB Manager**.

Note For more information about the client ID, see Client ID.

The list below describes situations during which this may occur, and includes a resolution.

Multiple user accounts with the same client ID

In some cases, multiple computers may inadvertently use the same client ID. When this occurs and computers with the same client ID attempt to connect with the same Hub, the first computer to associate itself with the Hub will be able to connect to the Hub. Subsequent computers will not be able to connect that Hub.

Resolution

If you discover that multiple computers are assigned the same client ID, see AnywhereUSB Manager client ID is not unique for help solving this issue.

AnywhereUSB Manager was uninstalled and then reinstalled

The **AnywhereUSB Manager** was completely removed from the PC, and then reinstalled. In this situation the **Manager** creates a new certificate for the client ID during the reinstall process.

Resolution

You can fix the client ID and Hub certificates mismatch by redeploying the Hub. See Deploy a Hub.

AnywhereUSB Manager created a new certificate

The **AnywhereUSB Manager** created a new certificate for some other reason, such as a factory reset of the **Manager**.

Resolution

You can fix the client ID and Hub certificates mismatch by redeploying the Hub. See Deploy a Hub.

Invalid client ID

The client ID is a unique identifier assigned to a user account the first time a user logs in to a computer and opens the **AnywhereUSB Manager**. The client ID is associated with the login credentials for the user currently logged on to the computer.

In some situations, the client ID is not registered with the Hub, and a connection between the Hub and the PC cannot be established. When this happens, the message "Invalid client ID" displays as the **State** in the **AnywhereUSB Manager**.

Note For more information about the client ID, see Client ID.

Client ID has not been added to the Hub

The client ID has not been added to the list of client IDs for the Hub.

Resolution

Add the client ID, which creates a certificate for the client ID.

- You can add a client ID to the Hub during the **AnywhereUSB Manager** installation process. See Client ID.
- You can manually add a client ID to the client list for the Hub. See Manually add a client ID.

Deploy a Hub

This process describes how to re-deploy a Hub.

Note If you want to deploy a new Hub, see Get started.

1. If you are re-deploying a Hub, remove the Hub from the network.
2. Make a connection between the Hub and each computer that should be allowed to connect to the Hub.
    a. Step 5: Connect to the device using an Ethernet LAN connection.
    b. Step 6: Verify initial connection.

Red X icon next to a Hub in the AnywhereUSB Manager

In some situations, a red X displays next to a Hub in the **AnywhereUSB Manager** when the Hub has failed to connect to your PC or the network. The list below describes situations during which this may occur, and includes a resolution.

Note If you do not want to display the Hubs that have failed to connect with your computer, you can hide them. See Hide all unauthorized Hubs.

Unable to connect

This status message displays in the **Hub Status** pane when the Hub is included in the known Hubs list but Hub is offline or the network is unreachable. For example, a firewall issue or other network issue could be blocking access from the Manager to the Hub.

TCP port is not configured correctly

The Hub cannot be reached via the TCP port (18574 by default) that is used by the **AnywhereUSB Manager** and is listened to by the Hub. Both the Hub and the **Manager** must be configured with the same TCP port in order for the Hub to connect to the client.

Resolution

Verify that the TCP port settings match for the Hub and the client.

- Hub: See AnywhereUSB Configuration page.
- Client: Verify the TCP port on your computer.

Hub is offline

The Hub could be powered off.

Resolution

Verify that it is connected to a power source and turned on.

Invalid Hub certificate

In some situations, the Hub certificate may become invalid. The Hub and the **AnywhereUSB Manager** must have matching certificates to be able to communicate. If the certificates do no match, the Hub and the **AnywhereUSB Manager** cannot communicate and a red X displays next to the Hub name in the **Manager**.

Resolution

For more information, see Manage Hub credentials and Invalid Hub certificate.

Hub has a different IP address

The device is no longer connected or has been moved to another network segment. The **AnywhereUSB Manager** does not discover Hubs that are not on the same network segment as the client.

Resolution

Add the Hub to the list of known Hubs. This ensures that the **AnywhereUSB Manager** can connect to the Hub, even it is on a different network. See Manage the list of known Hubs.

Note If you add a Hub to the list of known Hubs and you have the Hub autofind feature enabled, this may result in a duplicate connection for the same Hub. See Duplicate Hub, below.

Network issue blocking access

You should verify whether a network issue is blocking access to the Hub.

Attempt to ping the Hub:

- If you have a firewall that blocks TCP ports but allows ping, you will see successful pings but still not be able to connect. Contact your system administrator to verify that your firewall is not blocking TCP ports.
- If you can ping the Hub and are able to connect, a network issue does not exist and a different issue has occurred.
- If you cannot ping the Hub, check the configuration of the PC, and the Hub network settings, including firewalls and the network between them.

Duplicate Hub

If you have added a Hub to the known Hub list that is on same network as your computer, and you have the **Autofind Hubs** feature enabled, the Hub is found twice. The **AnywhereUSB Manager** attempts both connections, and the first one to connect will connect as expected. The second connection is discovered as a duplicate, and the **Manager** closes that connection and red X displays.

For more information, see Working with the known Hubs list and the Autofind Hubs option.

Resolution

The Hub added to the known Hubs list is considered a duplicate Hub, and should be removed from the known Hubs list.

Old version of AnywhereUSB Manager

In same cases, a Hub cannot connect to an older version of the **AnywhereUSB Manager**.

Resolution

Update to the most recent version of the **AnywhereUSB Manager**. See Step 3: Install the AnywhereUSB Manager.

Incompatible Hub

In some cases, the Hub firmware is old and must be updated to ensure that it can connect to the **AnywhereUSB Manager**.

Resolution

Update to the most recent version of the Hub firmware. See Update system firmware.

Unregistered Client ID

This status message displays in the **Hub Status** pane when the Hub is found on the network, but the unique client ID for the user logged into the computer is not registered with the Hub.

Initial connection

A red X displays next to a Hub name during the initial connection of the hardware to your PC. This is expected, and is a security feature.

For an example, see Step 6: Verify initial connection.

Resolution

The Hub administrator needs to allow each new client ID to connect to the Hub by adding the client ID to the client list. See Configure and manage client IDs.

Cannot uninstall the Manager from the Windows Apps screen

In some situations the **Modify** and **Uninstall** buttons in the Windows **Settings** > **Apps** > **Apps & Features** screen are both gray and do not activate when pressed. In this situation you must uninstall the Manager using from the Windows Control Panel.

Note You can also uninstall the **AnywhereUSB Manager** using the **AnywhereUSB Manager** installer. See Uninstall the AnywhereUSB Manager.

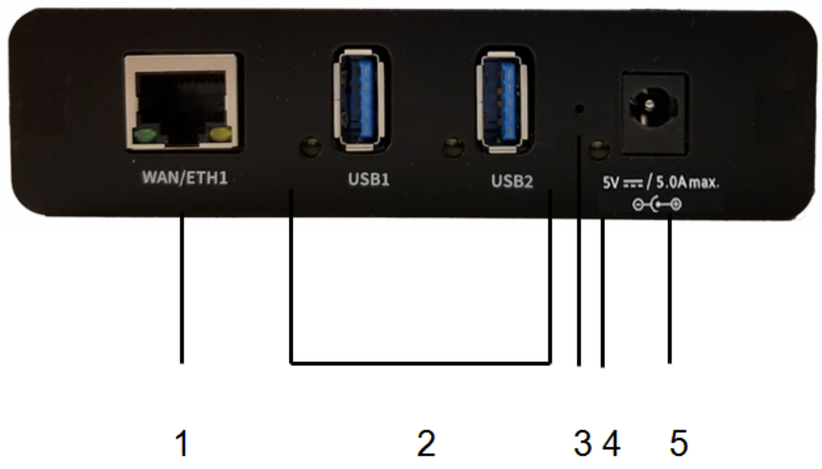
To uninstall the **Manager** from the Windows Control Panel:

1. Open the Control Panel and select **Progams and Features**.
2. Select **Digi AnywhererUSB Manager** from the list.
3. Click **Change**. You may have to right-click on **Digi AnywhererUSB Manager** to see the option. The **AnywhereUSB Manager** installation wizard appears.
4. Click **Next**. The **Program Maintenance** window appears.
5. Select the **Remove** option.
6. Click **Next**. The **Remove the Program** screen appears.
7. Make sure that **Remove User Configuration** is not selected. This preserves your current configuration.
8. Click **Remove**.

Hardware

The physical dimensions, environmental, and power requirements of the AnywhereUSB Hub can be found in the AnywhereUSB Plus datasheet.

AnywhereUSB 2 Plus: Front panel



Item	Name	Description
1	Ethernet connector	Connect the STP Cat 7 Ethernet cable.
2	USB LEDs and ports	The USB port supports 1.1, 2.0, and 3.1 USB devices. The LED illuminates as follows, based on the speed of the USB device: ■ 1.1 (Full speed): Yellow ■ 2.0 (High speed): Green ■ 3.1 (Super speed): Blue
3	Reset button	Use this button to reset the AnywhereUSB Hub configuration to factory defaults. See Erase device configuration and reset to factory defaults.
4	(Power) LED	The LED is solid blue when the device is powered on. This LED is also used for the Find Me feature. When this feature is activated, the LED blinks green and then orange.
5	Power connector	Connect the power supply: 5 Volt DC center positive. The Hub draws 5 Amp maximum when both USB ports are drawing 1.8 Amps each.

Attach a DIN rail clip (AnywhereUSB Plus 2-port ONLY)

Note You can attach a DIN rail clip only to a AnywhereUSB Plus 2-port device.

Before you begin

- You must purchase a DIN rail mounting kit: Digi PN 7000682.

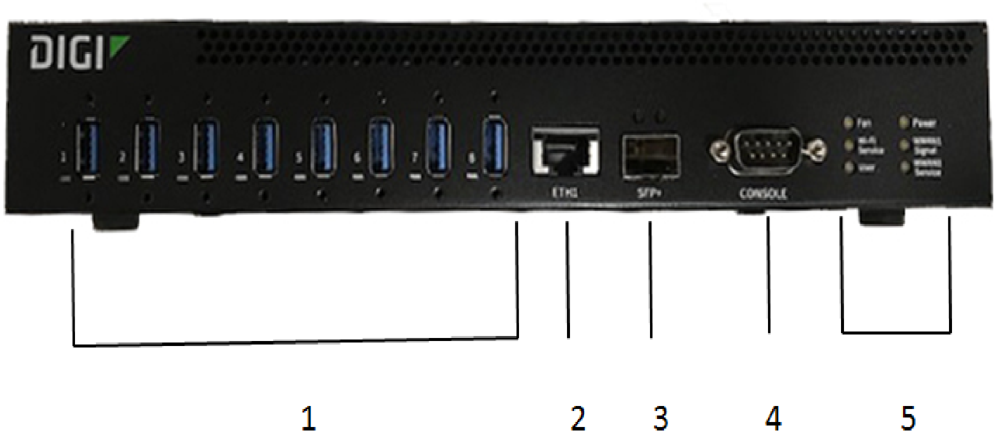
  Note Some kits may not have the required screws included. If this occurs, you will need to separately purchase two screws of the following type: 4-40 x .250 Flat head, Phillips head, zinc-plated screws.

- You will need an appropriate Phillips-head screwdriver.

Attach the DIN rail clip to the device

1. Remove required items from DIN rail mounting kit:
   - DIN rail clip
   - Two 4-40 x .250 Flat head, zinc-plated screws.
2. Place the DIN rail clip on the rear panel of the device. Make sure the screw holes are aligned.
3. In each hole, use a Phillips-head screw driver to screw in a 4-40 x .250 Flat head, zinc-plated screw.
4. Tighten the screws as needed to securely fasten the DIN rail clip to the device.
5. Use the DIN rail clip to mount the device to a rail.

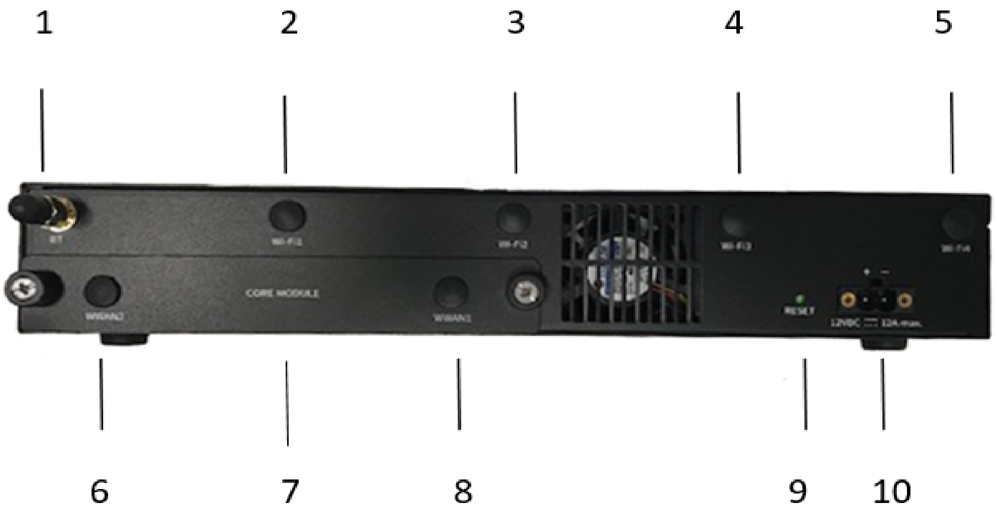
AnywhereUSB 8 Plus: Front panel



Item	Name	Description
1	USB ports and LEDs	The USB ports support 1.1, 2.0, and 3.1 USB devices. The LED illuminates as follows, based on the speed of the USB device: ■ 1.1 (Full speed): Yellow ■ 2.0 (High speed): Green ■ 3.1 (Super speed): Blue
2	ETH1	Connect a Cat 7 STP Ethernet cable. See Step 5: Connect to the device using an Ethernet LAN connection. **Note** Digi recommends that you use either the Ethernet cable or the SFP+ module. If both the Ethernet cable and the SFP+ module are connected, the SFP+ module will have priority.
3	SFP+	Connect an SFP transceiver module for fiber connection, such as Finisar Network FTLX8574D3BCL SFP+.
4	DB9 console	Used to access a console using the RS232 DTE interface. See
5	Fan	The LED shows the status of the fan:
		🟩 **Solid green** Fan is running within normal range of use

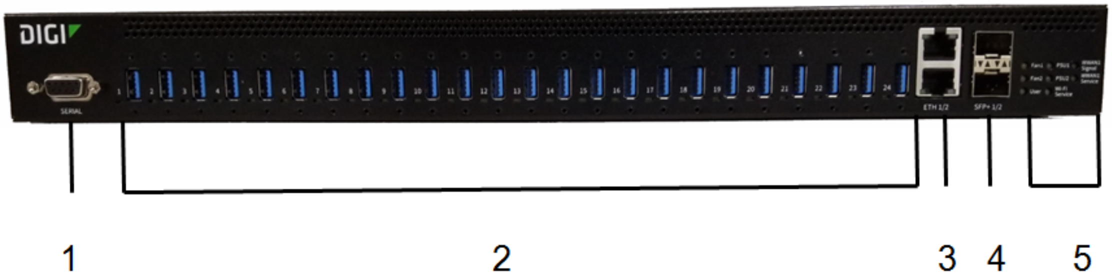
Item	Name	Description	
		🟥	**Solid red** Fan slows down or the Hub is overheating
5	Wi-Fi Service	Reserved for future use.	
5	User	LED used for the Find Me feature. When this feature is activated, the LED blinks orange and then green.	
5	Power	🟦	**Solid blue** The Hub is powered on.
5	WWAN1 Signal	🟥	**Solid red** Very poor signal (-113 dBm to -99 dBm)
		🟧	**Solid orange** Poor signal (-98 dBm to -87 dBm)
		🟨	**Solid yellow** Fair signal (-86 dBm to -76 dBm)
		🟩	**Solid light green** Good signal (-75 dBm to -64 dBm)
		🟩	**Solid green** Excellent signal. (-63 dBm to -51 dBm)
5	WWAN1 Service		**Off** No cellular service
		🟨⬜	**Flashing yellow** Cellular connection coming up
		🟨	**Solid yellow** Connected to 2G or 3G
		🟩	**Solid green** Connected to 4G

AnywhereUSB 8 Plus: Back panel



Item	Name	Description
1	BT	Reserved for future use.
2	Wi-Fi1	Reserved for future use.
3	Wi-Fi2	Reserved for future use.
4	Wi-Fi3	Reserved for future use.
5	Wi-Fi4	Reserved for future use.
6	WWAN2	Attach a cellular module antenna.
7	CORE module	Insert a CORE module component.
8	WWAN1	Attach a cellular module antenna.
9	Reset button	Use this button to reset the AnywhereUSB Hub configuration to factory defaults. See Erase device configuration and reset to factory defaults.
10	Power connector	Connect the power supply. See Step 5: Connect to the device using an Ethernet LAN connection.

AnywhereUSB 24 Plus: Front Panel

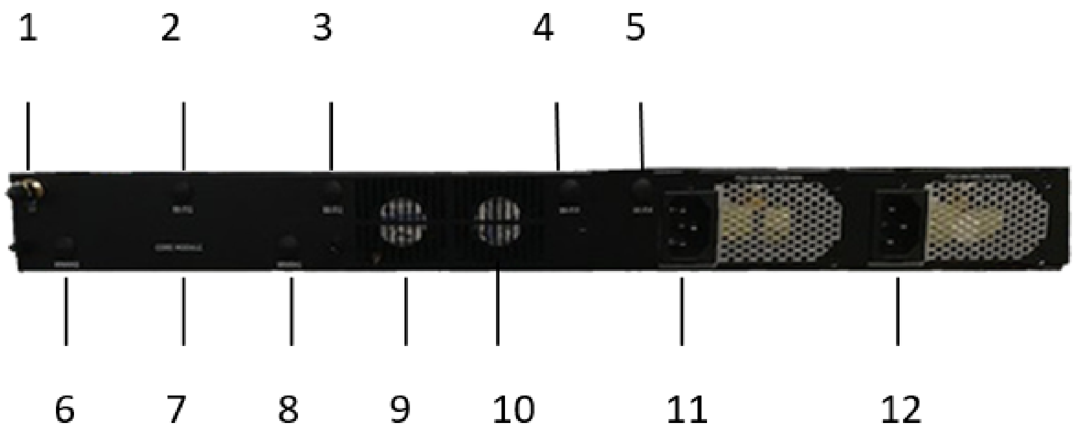


Item	Name		Description
1	DB9 Console	Used to access a console using the RS232 DTE interface.	
2	USB port LEDs	The USB ports support 1.1, 2.0, and 3.1 USB devices. The LED illuminates as follows, based on the speed of the USB device: ■ 1.1 (Full speed): Yellow ■ 2.0 (High speed): Green ■ 3.1 (Super speed): Blue	
3	ETH 1/2	Connect a Cat 7 STP Ethernet cable. ETH1 is the primary network interface. See Step 5: Connect to the device using an Ethernet LAN connection. ETH2 is the secondary network interface. This is optional and used for redundancy. **Note** Digi recommends that you use either the Ethernet cable or the SFP+ module. If both the Ethernet cable and the SFP+ module are connected, the SFP+ module will have priority.	
4	SFP+ 1/2	Connect an SFP transceiver module for fiber connection, such as Finisar Network FTLX8574D3BCL SFP+. The second SFP+ module is optional and used for redundancy.	
5	Fan1	The LED shows the status of Fan 1:	
		🟩	**Solid green** The fan is running within normal range of use.
		🟥	**Solid red** The fan slows down or the Hub is overheating.

Item	Name		Description
5	Fan2	The LED shows the status of Fan 2:	
		🟩	**Solid green** The fan is running within normal range of use.
		🟥	**Solid red** The fan slows down or the Hub is overheating.
5	User	LED used for the Find Me feature. When this feature is activated, the LED blinks orange and then green.	
5	PSU1	🟦	**Solid blue** The Hub is powered on.
		🟥	**Solid red** The Hub is not powered or the supply has failed.
5	PSU2	🟦	**Solid blue** The Hub is powered on.
		🟥	**Solid red** The Hub is not powered or the supply has failed.
5	Wi-Fi Service		Reserved for future use.
5	WWAN1 Signal	🟥	**Solid red** Very poor signal (-113 dBm to -99 dBm)
		🟧	**Solid orange** Poor signal (-98 dBm to -87 dBm)
		🟨	**Solid yellow** Fair signal (-86 dBm to -76 dBm)
		🟩	**Solid light green** Good signal (-75 dBm to -64 dBm)
		🟩	**Solid green** Excellent signal. (-63 dBm to -51 dBm)

Item	Name		Description
5	WWAN1 Service		**Off** No cellular service
		🟨⬜	**Flashing yellow** Cellular connection coming up
		🟨	**Solid yellow** Connected to 2G or 3G
		🟩	**Solid green** Connected to 4G

AnywhereUSB 24 Plus: Back panel



Item	Name	Description
1	BT	Reserved for future use.
2	Wi-Fi1	Reserved for future use.
3	Wi-Fi2	Reserved for future use.
4	Wi-Fi3	Reserved for future use.
5	Wi-Fi4	Reserved for future use.
6	WWAN2	Attach a cellular module antenna.
7	CORE module	Insert a CORE module component.
8	WWAN1	Attach a cellular module antenna.
9	Fan 1	Primary fan.
10	Fan 2	Secondary fan.
11	Power connector	Connect the power supply. See Step 5: Connect to the device using an Ethernet LAN connection.
12	Power connector	Connect the second (optional) power supply. This is used for redundancy.
	Reset button	The reset button is on the side of the Hub. Press this button to reset the AnywhereUSB Hub configuration to factory defaults. See Erase device configuration and reset to factory defaults.

QR code definition

A QR code is printed on the label attached to the device and on the loose label included in the box with the device components. The QR code contains information about the device.

QR code items

Semicolon separated list of:

ProductName;DeviceID;Password;SerialNumber;SKUPartNumber SKUPartRevision

Note There is a space between PartNumber and PartRevision.

Example

AnywhereUSB 8 Plus;00000000-00000000-112233FF-FF445566;PW1234567890;AW08-123456;AW08-G300 E

Regulatory and safety information

Safety warnings

Review the following safety warnings for AnywhereUSB Hub.

WARNING! Notice the following safety warnings:

- Risk of explosion if battery is replaced by incorrect battery type. Dispose of used batteries according to the instructions.
- This equipment is not suitable for use in locations where children are likely to be present.
- This is equipment is suitable for installation in Information Technology Rooms in accordance with Article 645 of the National Electrical Code and NFPA 75.
- Use certified and rated Laser Class I Optical Transceiver product.
- Ensure that the power cord is connected to a socket-outlet with earthing connection.
- Never open the equipment. For safety reasons, the equipment should be opened only by skilled person.

WARNING! Risk of electric shock.

WARNING! Disconnect all energy sources.

AVERTISSEMENT! Notice the following safety warnings:

- Risque d'explosion si la batterie est remplacée par un type de pile incorrect. Jeter les piles usées selon les instructions.
- Cet équipement ne convient pas pour une utilisation dans des endroits où des enfants sont susceptibles d'être présents.
- Convient pour l'installation dans les salles informatiques conformément à l'article 645 du code national de l'électricité et à la norme NFPA 75.
- Utilisez un produit émetteur-récepteur optique laser de classe I certifié et évalué.
- Veillez à connecter le cordon d'alimentation à une prise de courant avec mise à la terre.
- Ne jamais ouvrir l'équipement. Pour des raisons de sécurité, l'équipement ne doit être ouvert que par du personnel qualifié.

AVERTISSEMENT! Attention! Danger de choc.

AVERTISSEMENT! Déconnecter toutes les source d'énergie.