Link-OS

PrintSecure

Printer Administration Guide

ZEBRA

P1099957-07EN

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: zebra.com/informationpolicy
COPYRIGHTS: zebra.com/copyright
WARRANTY: zebra.com/warranty
END USER LICENSE AGREEMENT: zebra.com/eula

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Publication Date

May 11, 2022

Table of Contents

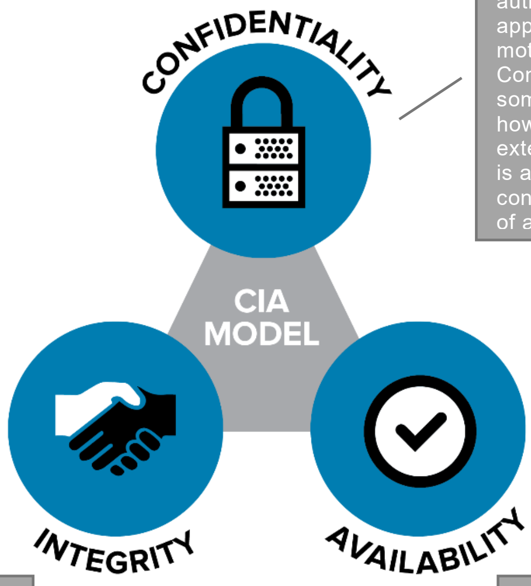
Introduction

This document details how to perform Administrator-level functions on a Zebra Label or Receipt printer. The content in this document covers both Link-OS® and ZebraLink™ printers, though the degree to which the two types of printers can be Administered is different. To make it easy to see where a given Administrative feature is available, the document will display the Link-OS or Zebra Link™ ZebraLink icon to indicate if the feature is available on the printer being configured.

Overview

Administering Thermal label and receipt printers might, at first, appear to be a very different task than managing other devices, such as computers or smartphones. Fortunately, there is a well-established, reliable model and a set of best practices that can be easily applied to minimize risks and make the task straightforward.

The "CIA Model" provides a guiding framework when considering how to reasonably and effectively raise the bar on risk mitigation. The model can be applied to all devices that utilize the data protected by enterprise information systems, from the more traditional connected solutions to the new players in the connected environment, such as intelligent thermal barcode printers. It includes three components:



The concept of **confidentiality** is to ensure that information is only available to the people who are authorized to access it. This protection applies equally to data at rest, in motion and during processing. Confidentiality and Privacy are sometimes used interchangeably; however, confidentiality is normally an extension of privacy. Data encryption is a common method of ensuring confidentiality as are various methods of authentication for authorization.

The Concept of data **integrity** is to ensure consistency, accuracy and trustworthiness of the data over its entire lifecycle. It also means that when a file is stored and then accessed later, there will be certainty that the data had not been directly or indirectly altered by an unauthorized entity while in storage. Access controls can prevent unauthorized access and checksums (CRC), cryptography (Hashes) and Digital Signatures can be used to validate integrity.

The Concept of **availability** is to ensure that the resource/device is available when the user needs it. This can include keeping the resource/device online and rigorously updated to prevent attacks that affect the stability of the device. Some of the responsibility for availability also relies upon the host that the resource/device is connected to and also how it is connected (Wired/Wireless). Business critical resource/devices need to be available when needed, otherwise, a plan needs to be in place to replace the device and reduce downtime.

Common Sense Best Practices

There are a set of Best Practices you can put in place to align your printer Administration with the CIA concepts. By applying these common-sense Best Practices, you can reduce risk, while still optimizing your use of thermal barcode printers.

1

- Start early. Plan for incoming devices, and how you’ll protect them.

2

- Use encrypted and authenticated connections where possible.

3

- Plan to rotate access passwords, access keys and authentication credentials.

4

- Defaults typically represent documented methods to access a device. Activate User Interface Passwords and consider turning off the device services that you don’t plan to use.

5

- Leverage a remote management system to allow you to quickly update settings and standards. The longer devices are using out of date settings, the longer they represent the *easier target*.

6

- Keep update schedules and plans only in the hands of those who need to have them. Knowing when updates are planned can inadvertently encourage inappropriate actions.

7

- Plan for a method to continuously monitor your system for *out of touch* devices. Where you suspect a device has been taken out of your environment, withdraw its credentials until the device status is determined.

8

- Choose devices that can be updated across their long service lives so they keep current with new standards. Verify that the update system uses a method to ensure the update file hasn’t been tampered with.

9

- Plan for device retirement by removing enterprise system settings, deleting device user Accounts/Credentials and checking to make sure the existing system isn’t hardcoded to look for retired devices.

10

- Consider Confidentiality, Integrity, and Availability during all stages of the devices lifecycle.

Steps to Take:

Applying these Best Practices is straightforward. The process involves four steps:

1. **Census** – which devices do you have?
2. **Consider** – which Admin capabilities do your printers have?
3. **Configure** – send commands to alter Admin settings
4. **Confirm** – validate the new settings

Census: Which Devices Do You Have?

Zebra has been making printers for over 30 years. In that time, the scope of Administrator-level settings has grown. It is important to know which printer models you are working with to know which Administrator controls are available. The chart below will help you *place* your printer model into one of three categories:

Legacy Models (no admin features)	Zebra Link™ (limited admin features)	**Link-OS®** (most admin features)
Desktop Printers A100 series A300 series Bravo series Companion Encore series LP/TLP series Tiger Writer 2746 series HT146 DA402 R402 T300/T402	**Desktop Printers** LP/TLP-Z series LP/TLP Plus series S300 S400 S500 S600 G series HC100	**Desktop Printers** ZD200 series ZD400 series ZD500 series ZD600 series
Mobile Printers Cameo series MP series QL series PA400 series PT400 series PS2000-PS400 series TR220 ZQ110	**Mobile Printers** QLPlus series P4T series RW Series	**Mobile Printers** iMZ series (up to Link-OS 5.2) QLn series (up to Link-OS 5.2) ZQ300 series ZQ500 series ZQ600 series ZR300 series ZR600 series
Industrial Printers Z60 series Z90 series Z100 series Z140 series Z200 series 105Se	**Industrial Printers** Z4000/Z6000 Z4M/Z6M ZM400/600 series 105SL series 105SL Plus series XiII through Xi4 series	**Industrial Printers** ZT200 series ZT400 series ZT500 series ZT600 series
Others TTP Kiosk printer series	**Others** PAX 2 through PAX5 series ZE500 series KR403	**Others** ZE501 series

Consider: Which Admin Capabilities Does Your Printer Have?

Link-OS printers support a wide range of administrative commands and features.

	Zebra Link™	
Security		
Protected Mode		✓
OS Download Blocking		✓
Decommissioning Mode		✓
Services		
HTTP	✓	✓
HTTPS		✓
FTP	✓	✓
LPD	✓	✓
UDP		✓
SMTP	✓	✓
SNMP	✓	✓
Raw Telnet	✓	✓
POP3	✓	✓
NTP		✓
Communications		
Auto-WLAN Cert Management		✓
Bluetooth Mode		✓
Bluetooth Discoverability		✓
Bluetooth Enable		✓
BTLE		✓
USB Host		✓
Ethernet		✓
WLAN		✓
ESSID		✓
802.11x		✓
RTS/CTS Protection		✓
IP Address Allow list		✓
IP Port		✓
IP Alternate port		✓
JSON port		✓
Single connection port		✓
TLS IP Port		✓
TLS JSON Port		✓
TLS Enable		✓
Web sockets port		✓
Asset Visibility Agent		✓
MQTT		✓
Applications		
Data Capture		✓
XML Printing	✓	✓
USB Mirror		✓
FTP Mirror	✓	✓
SFTP Mirror		✓
Zebra Basic Interpreter		✓
APL Emulations		✓
User Interface		
Password	✓	✓

Premade Administration Files

Zebra has created several sets of pre-made files that you can send to your printer to quickly enable some of the most common security settings. These pre-made Admin Files were designed and built using the commands documented in this guide. However, because different user's networks operate in different ways, there is no one configuration file that could address every user's needs.

To obtain the pre-made Admin Files, go to: https://www.zebra.com/printsecure

You should edit the files to adapt to your unique needs. As you work with the Printer Administration Guide, you will quickly discover which commands and settings that are appropriate for your use case. For example, if your application uses Mirror, then turning off FTP would not make sense, since Mirror uses FTP to communicate to the printer. This example demonstrates why it is important to consider the following pages below before sending the files.

Sending the Administration files is simple. You can send the files to any port on the printer using our Printer Setup Utility or the legacy Z-Downloader utility.

The Printer Setup Utility can be downloaded from: www.zebra.com/setup

The legacy Z-Downloader app can be downloaded from:

https://www.zebra.com/us/en/support-downloads/printer-software/zdownloader.html

The Premade Administration files come in four groups:

- **Applications** – Three files, which can be used to set, check settings, or default the application settings on the printer.
- **Communications** – Three files, which can be used to set, check settings, or default the communication settings on the printer.
- **Services** – Three files, which can be used to set, check settings, or default the services settings on the printer.
- **User interface** – Two files, which can be used to set or default the user interface settings on the printer. (Important note: Zebra recommends that to not use the sample password shown in this file, please change it.)

Configure:

Send Commands to Alter Admin Settings

Confirm:

Validate the New Settings

This can be the most time-consuming portion of the process. Each Administrative capability used will have consequences for how the printer works, what it can do, and how it will work with other devices. Time should be taken to carefully consider which administrative features are used, and how they may impact the use of the printer.

Commands:

In this section, each Admin capability will be detailed, along with its defaults, its range of settings, how to activate/deactivate it, along with some notes to help you carefully consider the use of the capability.

NOTE: Many of the Administrative capabilities are controlled using the Set-Get-Do command language. If you are not familiar with this language, please consult the Zebra Programming Guide, SGD Chapter for help with syntax and how to use this printer feature.

Protected Mode Commands

Services and Networking Commands

Communications Commands

Applications Commands

User Interface

Protected Mode Commands

<table>
<tr><th>PROTECTED MODE STATE</th><th colspan="2">Supported Printer Types</th></tr>
<tr><td><b>Description:</b> This command returns the current state of Protected Mode.</td><td></td><td></td></tr>
<tr><td colspan="3"><b>Considerations:</b> By default, Protected Mode is off.
It is recommended to place the printer into Protected Mode to prevent unintentional or unauthorized setting changes.</td></tr>
<tr><td colspan="3"><b>Control Commands:</b> Protected Mode is controlled by JSON commands.
This SGD command will report if Protected Mode is on or off. More detail can be found in the Best Practices - Protected Mode section of this guide.

Example:

    ! U1 getvar "device.protected_mode"

The printer responds with the current setting value: "on" or "off".

Return to Command List</td></tr>
</table>

Protected Mode Commands

PROTECTED MODE ALLOWED	Supported Printer Types	
Description: This command returns the state of Protected Mode Allowed. This is used in conjunction with setting the password.		

Considerations: It is recommended to place the printer into Protected Mode to prevent unintentional or unauthorized setting changes.

Control Commands: Protected mode is controlled by JSON commands and this SGD command will report if protected mode is allowed. More detail can be found in the Best Practices - Protected Mode section of this guide.

Example:

```
     ! U1 getvar "device.protected_mode_allowed"
```

The printer responds with the current setting value: "yes" or "no".

Return to Command List

Protected Mode Commands

PRINTER OS DOWNLOAD CONTROL	**Supported Printer Types**	
Description: This command controls the device firmware download capability.		

Considerations: The default for this setting is "yes". It is recommended that Printer OS Download control be enabled to prevent unplanned Printer OS updates. Protected Mode should also be enabled to protect this setting and prevent it from being altered.

Control Commands: The Printer OS Download Control capability is controlled by the `device.allow_firmware_downloads` command. More detail can be found in the Best Practices - Firmware Protection section of this guide.

To set the command:

```
! U1 setvar "device.allow_firmware_downloads" "yes"
! U1 setvar "device.allow_firmware_downloads" "no"
```

To confirm the command is set:

```
! U1 getvar "device.allow_firmware_downloads"
```

The printer responds with the current setting value: "yes" or "no".

To reset the device to the default state:

```
! U1 setvar "device.allow_firmware_downloads" "yes"
```

Return to Command List

Note: If this setting is set to "no", Printer OS downloads will not be possible. In this case the allow-next-firmware-download operation can be used to allow the next firmware file to be accepted. Please refer to the section labeled Printer OS Download Protection later in the guide for details.

Services and Networking Commands

HTTP SERVICE	Supported Printer Types	
Description: This service is used to provide HTTP access to the printer	Zebra Link™	

Considerations: The HTTP service runs on port 80 and provides support for the printer's internal web pages. It is important to note that any POST to URL capability is disabled when this service is not enabled. The printer can still be managed by the Printer Profile Manager Enterprise app or via direct commands when this is disabled.

Note: Alerts with this destination will not work when this service is disabled.

Control Commands: The HTTP capability is controlled by the ip.http.enable command

To set the command:

```
! U1 setvar "ip.http.enable" "on"
! U1 setvar "ip.http.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.http.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.http.enable" "on"
```

Return to Command List

Services and Networking Commands

HTTPS SERVICE	Supported Printer Types	
Description: This service is used to provide HTTPS access to the printer		

Considerations: The HTTPS service runs on port 443 and provides support for the printer's internal web pages utilizing a secure connection.

Control Commands: The HTTPS capability is controlled by the ip.https.enable command

To set the command:

```
    ! U1 setvar "ip.https.enable" "on"
    ! U1 setvar "ip.https.enable" "off"
```

To confirm the command is set:

```
    ! U1 getvar "ip.https.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "ip.https.enable" "on"
```

Return to Command List

Note: This command requires that a valid certificate is present on the printer.

The certificate and private key can be deployed to the device as a single file, or separate files. If using a single file, the name of the file must be:

HTTPS_CERT.NRD

If using multiple files:

HTTPS_CERT.NRD – certificate file

HTTPS_KEY.NRD – private key file

Once TLS communication is verified and operational, it is a security best practice to disable unencrypted forms of communicating with the printer over a network.

Services and Networking Commands

FTP SERVICE	Supported Printer Types	
Description: This service is used to send commands or files that the printer will act upon (this can include, CPCL, EPL, ZPL and Set-Get-Do commands).	Zebra Link™	

Considerations: The FTP service run on port 21 and can be used to place files on the printers file system, or for printing. It is also the protocol used by the Mirror device management features. It is not a service that is typically used for printing. As such, it’s a good candidate to be disabled, however, it’s important to first check if your organization plans to use it for file transfer, printing or device management.

Control Commands: The FTP capability is controlled by the "ip.ftp.enable" command

To set the command:

```
! U1 setvar "ip.ftp.enable" "on"
! U1 setvar "ip.ftp.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.ftp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.ftp.enable" "on"
```

Return to Command List

Note: Only Link-OS printers can use SFTP. For further information on FTP and SFTP Mirror, refer to the Programming Guide.

Services and Networking Commands

LPD SERVICE	Supported Printer Types	
Description: This service is used to send print jobs to the printer that it will act upon (this can include, CPCL, EPL, ZPL).	Zebra Link™	

Considerations: The LPD service uses port 515 and is a printing protocol typically used in Unix/Linux systems and the Mac OS environment. This can be supported on a Windows network with the addition of software features. Check which printing technology you are using and disable the appropriate port(s).

Control Commands: The LPD capability is controlled by the `ip.lpd.enable` command

To set the command:

```
! U1 setvar "ip.lpd.enable" "on"
! U1 setvar "ip.lpd.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.lpd.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.lpd.enable" "on"
```

Return to Command List

Services and Networking Commands

UDP SERVICE	Supported Printer Types	
Description: The UDP socket is only used for port defined by ip.port.		

Considerations: The User Datagram Protocol (UDP) is a connectionless protocol in contrast to Transmission Control Protocol (TCP) which requires a validated connection and an IP address. The primary purpose of this service is to communicate with the printer command language parser via UDP.

Note: Alerts with this destination will not work when this service is disabled. All other UDP based communication (such as SNMP) are not affected.

Control Commands: The UDP capability is controlled by the ip.udp.enable command

To set the command:

```
! U1 setvar "ip.udp.enable" "on"
! U1 setvar "ip.udp.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.udp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.udp.enable" "on"
```

Return to Command List

Services and Networking Commands

TCP SERVICE	Supported Printer Types	

Description: The TCP socket is used for ports defined by:
`ip.port`
`ip.port_alternate`
`ip.port_json_config`
`ip.port_single_conn`		

Considerations: The Transmission Control Protocol (TCP) provides a reliable, ordered, error-checked, validated connection in contrast to User Datagram Protocol (UDP). The primary purpose of this service is to communicate with the printer command language parser via TCP.

Note: Alerts with this destination will not work when this service is disabled. All other TCP based communication (such as HTTP) are not affected.

Control Commands: The TCP capability is controlled by the ip.tcp.enable command

To set the command:

```
! U1 setvar "ip.tcp.enable" "on"
! U1 setvar "ip.tcp.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.tcp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.tcp.enable" "on"
```

Return to Command List

Services and Networking Commands

SMTP SERVICE	Supported Printer Types	
Description: The Simple Mail Transfer Protocol (SMTP) service (port 25) is used to receive print jobs.	Zebra Link™	

Considerations: This SMTP service is used to receive printer jobs using the Simple Mail Transfer Protocol (this can include, CPCL, EPL, ZPL). The print job is sent in the body of the email. Please refer to the Zebra Printer Programming Guide for format.

Control Commands: The SMTP capability is controlled by the `ip.smtp.enable` command

To set the command:

```
! U1 setvar "ip.smtp.enable" "on"
! U1 setvar "ip.smtp.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.smtp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.smtp.enable" "on"
```

Return to Command List

Note: Ensure that the other dependent settings are configured correctly when using this capability

For further information on SMTP refer to the Programming Guide.

For example:

```
ip.smtp.server_addr
ip.smtp.domain
```

Services and Networking Commands

SNMP SERVICE	**Supported Printer Types**	
Description: The Simple Network Management Protocol (SNMP) service enables the manageability of the printer using this industry standard protocol.	Zebra Link™	

Considerations: The SNMP service uses UDP port 161 and allows the configuration of the printer and supports the issuance of SNMP trap messages. Some of the basic printer MIB is supported as well as a private MIB that contains Zebra specific settings and configuration. By default, this uses the public community name, if you intend to use this consider changing the community name from the default.

Note: Alerts with this destination will not work when this service is disabled.

Control Commands: The SNMP capability is controlled by the ip.snmp.enable command

To set the command:

```
     ! U1 setvar "ip.snmp.enable" "on"
     ! U1 setvar "ip.snmp.enable" "off"
```

To confirm the command is set:

```
     ! U1 getvar "ip.snmp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
     ! U1 setvar "ip.snmp.enable" "on"
```

Return to Command List

POP3 MAIL SERVICE	Supported Printer Types	
Description: The printer has a pop3 mail service and can poll a mailbox for incoming emails.	Zebra Link™	

Considerations: The POP3 service can query a mailbox for incoming emails, which can contain ZPL/CPL/EPL in the body of the email. The printer will execute the command language.

Note: Alerts with this destination will not work when this service is disabled.

Control Commands: The POP3 capability is controlled by the ip.pop3.enable command

To set the command:

```
! U1 setvar "ip.pop3.enable" "on"
! U1 setvar "ip.pop3.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "ip.pop3.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.pop3.enable" "on"
```

Return to Command List

Note: Ensure that the other dependent settings are configured correctly when using this capability

For further information on POP3 refer to the Programming Guide.

For example:

```
ip.pop3.server_addr
ip.pop3.poll
ip.pop3.username
ip.pop3.password
```

NTP SERVICE	Supported Printer Types	
Description: This command enables or disables the Network Time Protocol (NTP) feature.		

Considerations: The NTP command will enable or disable the Network Time Protocol capability which allows the printer to synchronize with time servers. This may be important if there are date or time fields printed on the label. Time and data can also be provided by the host system.

Control Commands: The NTP capability is controlled by the ip.ntp.enable command

To set the command:

```
    ! U1 setvar "ip.ntp.enable" "on"
    ! U1 setvar "ip.ntp.enable" "off"
```

To confirm the command is set:

```
    ! U1 getvar "ip.ntp.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "ip.ntp.enable" "off"
```

Return to Command List

Note: Ensure that the other dependent settings are configured correctly when using this capability

For further information on NTP refer to the Programming Guide.

For example:

```
ip.ntp.servers
ip.ntp.log
```

Services and Networking Commands

TIME	Supported Printer Types	
Description: This command sets or gets the printer time based on the Unix Epoch (UTC) or number of seconds since January 1st 1970.		
Considerations: If NTP is unavailable, time can be set using this command. Setting time in this way is useful for devices that exists across multiple time zones.		

Control Commands: The Unix Epoch capability is controlled by the rtc.unix_timestamp command

To set the command:

```
! U1 setvar "rtc.unix_timestamp" "1561492746" (06/25/2019 7:59PM (UTC))
```

To confirm the command is set:

```
! U1 getvar "rtc.unix_timestamp"
```

The printer responds with the current setting value in seconds.

Return to Command List

Note: The printer time and date can also be set using

```
rtc.time
rtc.date
```

It is possible to interrogate the printer to see if a real time clock chip is installed.

```
rtc.exists
```

Communications Commands

BLUETOOTH ENABLE	Supported Printer Types	
Description: This command enables or disables the Bluetooth radio in a printer that has that option installed.	Zebra Link™	

Considerations: If you utilize Bluetooth for connection to a mobile computer for printing, this will need to be configured correctly.

Note: Alerts with this destination will not work when this service is disabled.

Control Commands: The Bluetooth enable capability is controlled by the bluetooth.enable command

To set the command:

```
! U1 setvar "bluetooth.enable" "on"
! U1 setvar "bluetooth.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "bluetooth.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "bluetooth.enable" "on"
```

Return to Command List

Note: Please review changes to the default value of `bluetooth.discoverable` implemented in Link-OS 6.

Communications Commands

BLUETOOTH DISCOVERABILITY	**Supported Printer Types**	
Description: This command enables or disables the Bluetooth discoverable mode in a printer that has a BT option installed.	Zebra Link™	

Considerations: The Bluetooth discoverable command will disable the Bluetooth connectivity on the printer. This does not affect a previously paired device only the discovery and pairing of a new device.

Control Commands: The Bluetooth discoverable capability is controlled by the bluetooth.discoverable command

To set the command:

```
! U1 setvar "bluetooth.discoverable" "on"
! U1 setvar "bluetooth.discoverable" "off"
```

To confirm the command is set:

```
! U1 getvar "bluetooth.discoverable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "bluetooth.discoverable" "off"
```

Return to Command List

Note: The default value of this setting has changed as of Link-OS 6 and is now off by default to improve security. Bluetooth Discovery and Pairing Mode can be activated by holding the FEED button on the printer for 5 seconds. For further details please refer to the Link-OS 6 Release notes.

Communications Commands

BLUETOOTH MODE	Supported Printer Types	
Description: For printers that support both Bluetooth Classic and Bluetooth Low Energy (BTLE), this command controls the mode of operation.		

Considerations: The Bluetooth radio can be configured to work in the following mode; BTLE, Classic or Both.

Control Commands: The Bluetooth controller mode is controlled by the bluetooth.le.contoller_mode command

To set the command:

```
    ! U1 setvar "bluetooth.le.contoller_mode" "both"
    ! U1 setvar "bluetooth.le.contoller_mode" "le"
    ! U1 setvar "bluetooth.le.contoller_mode" "classic"
```

To confirm the command is set:

```
    ! U1 getvar "bluetooth.le.contoller_mode"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "bluetooth.le.contoller_mode" "both"
```

Return to Command List

Note: There are many other settings related to BT communication and these need to be reviewed and configured accordingly.

Please review changes to the default value of `bluetooth.discoverable` implemented in Link-OS 6.

For further information on Bluetooth refer to the Programming Guide.

For example:

```
    bluetooth.discoverable
    bluetooth.minimum_security_mode
    bluetooth.allow_non_display_numeric_comparison
    bluetooth.bonding
    bluetooth.pin
```

Communications Commands

USB HOST	Supported Printer Types	
Description: This command is used to enable or disable USB host capabilities in a printer that supports USB Host		

Considerations: The USB host lockout command disables the USB host capability in a printer that has support for it. USB devices connected to the printer will stop functioning when this is disabled. This will include USB mirror if that is being used.

Control Commands: The USB host lock out capability is controlled by the usb.host.lock_out command

To set the command:

```
    ! U1 setvar "usb.host.lock_out" "on"
    ! U1 setvar "usb.host.lock_out" "off"
```

To confirm the command is set:

```
    ! U1 getvar "usb.host.lock_out"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "usb.host.lock_out" "off"
```

Return to Command List

WIRED ETHERNET	Supported Printer Types	
Description: Enable or disable the internal wired ethernet port on printers equipped with this option.		

Considerations: The wired LAN enable command will disable or enable the internal wired Ethernet connection. The primary use for this command is to disable a port that is unused, where a different port is being used as the primary connection.

Control Commands: The wired LAN capability is controlled by the internal_wired.enable command

To set the command:

```
! U1 setvar "internal_wired.enable" "on"
! U1 setvar "internal_wired.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "internal_wired.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "internal_wired.enable" "on"
```

Return to Command List

Communications Commands

WLAN	Supported Printer Types	
Description: This command is used to enable or disable the Wireless Local Area Network functionality in a printer equipped with the WLAN (Wi-Fi) option.		

Considerations: The default value for `wlan.enable` is "on".
The WLAN command will fully disable all 802.11 wireless functionality. To improve security, it is recommended that the value of `wlan.enable` be set to "no" if the WLAN (Wi-Fi) option is not being used.

Control Commands: The WLAN capability is controlled by the `wlan.enable` command

To set the command:

```
! U1 setvar "wlan.enable" "on"
! U1 setvar "wlan.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "wlan.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "wlan.enable" "on"
```

Return to Command List

Communications Commands

ESSID	Supported Printer Types	
Description: This command is used to configure the WLAN Extended Service Set Identifier (ESSID) value, which determines which Wireless Local Area Network the device will connect to.		

Considerations: Set the ESSID network name to match the value of the WLAN the device will connect to automatically. The default value for ESSID is "" (null), which prevents the device from associating to any Access Point.

Control Commands: The WLAN network name is controlled by the `wlan.essid` command

To set the command:

```
     ! U1 setvar "wlan.essid" "networkName"
```

To confirm the command is set:

```
     ! U1 getvar "wlan.essid"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
     ! U1 setvar "wlan.essid" ""
```

Return to Command List

Note: For versions prior to Link-OS 6, the default value for ESSID is "`125`". This allowed device administrators to create a network specifically for provisioning new devices quickly. If the device ESSID is set to "" (null), the device will attempt to associate to any available Access Point, regardless of what its ESSID value is.

In Link-OS 6 and higher, the device will not automatically associate to any Access Point until a valid ESSID value is set.

Communications Commands

WIRELESS OPTION	Supported Printer Types	
Description: This option provides a mechanism to authenticate devices on a LAN		
Considerations: When using the 802.1x authentication user must be aware of the movement of data to the printer during setup. Best practices should be employed to ensure that certificates and passphrases are protected at all time. Configuration should be done over a local connection to prevent eavesdropping.		

Control Commands:

To set the command:

```
! U1 setvar "wlan.8021x.enable" "on"
! U1 setvar "wlan.8021x.enable" "off"
! U1 setvar "wlan.8021x.enable" "wpa"
```

To confirm the command is set:

```
! U1 getvar "wlan.8021x.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "wlan.8021x.enable" "off"
```

Return to Command List

Note: There are many other settings related to 802.1x Authentication and these need to be reviewed and configured accordingly.

For further information on 802.1x refer to the Programming Guide.

For example:

```
wlan.8021x.authentication
wlan.8021x.ttls_tunnel
wlan.8021x.peap.peap_username
wlan.8021x.peap.peap_password wlan.8021x.peap.privkey_password
wlan.8021x.peap.validate_server_certificate
wlan.8021x.peap.anonymous_identity
wlan.8021x.eap.username
wlan.8021x.eap.password
wlan.8021x.eap.privkey_password
```

Communications Commands

WLAN RTS/CTS	Supported Printer Types	
Description: Enables RTS/CTS HT protection frames when configuring a WLAN connection.	Zebra Link™	

Considerations: The WLAN RTS_CTS feature when enabled will put the WLAN radio in RTS/CTS protection mode. If this is not enabled the radio will default to CTS-to-Self mode. The mode that you run in will be dependent on your specific wireless LAN configuration and the devices that connect to it.

Control Commands: The WLAN RTS_CTS capability is controlled by the wlan.rts_cts_enable command

To set the command:

```
! U1 setvar "wlan.rts_cts_enabled" "on"
! U1 setvar "wlan.rts_cts_enabled" "off"
```

To confirm the command is set:

```
! U1 getvar "wlan.rts_cts_enabled"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "wlan.rts_cts_enabled" "off"
```

Return to Command List

Note: This command functions on the QLn and ZQ500 series printers.

Communications Commands

ALLOW LISTING	Supported Printer Types	
Description: The allow list capability allows only authorized IP addresses to connect to the printer.		

Considerations: The allow list capability is to ensure that only authorized hosts can connect to the printer. The parameters that you set are the IP addresses that are permitted to connect and can be single IP address or ranges. The maximum string length allowed is 256 bytes.

Control Commands: The allow list capability is controlled by the `ip.firewall.whitelist_in` command.

To set the command:

```
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20"
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20, 192.168.100.21"
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20-192.168.1.100"
```

To confirm the command is set:

```
! U1 getvar "ip.firewall.whitelist_in"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.firewall.whitelist_in" ""
```

Return to Command List

Note: This command allows up to 256 characters that define what IP’s or ranges of IP’s can connect to the printer. If the IP address is not listed the connection will be refused. To reset this list, you will need to connect to a local port and send this command if the IP you are trying to connect with is not in the allowed range.

Examples:
Single IP address

```
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20"
```

Multiple IP addresses

```
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20,192.168.1.21"
```

IP address ranges

```
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20-192.168.1.40"
```

IP ranges and Single/Multiple IPs

```
! U1 setvar "ip.firewall.whitelist_in" "192.168.1.20-192.168.1.40,
192.168.1.50, 192.168.1.75"
```

Communications Commands

TCP RAW PORT	Supported Printer Types	
Description: This port is used to send commands or files that the printer will act upon (this can include, CPCL, EPL, ZPL and Set-Get-Do commands). This is also known as the printer command language parser.	Zebra Link™	
Considerations: Since this is frequently the primary port used for network-based printing, disabling it could disable printer. Of course, printing could be happening over another port, via FTP or web sockets. Additionally, changing the port number used could help obscure the printing port, but note that the most port scanning tools can easily discover which ports are open on a networked device.		

Control Commands: The TCP Raw Port setting is controlled by the "ip.port" command
To set the command:
`! U1 setvar "ip.port" "9100"`
`! U1 setvar "ip.port" "0" (Disables port)`
To confirm the command is set:
`! U1 getvar "ip.port"`
The printer responds with the current setting value, or "?" if not supported.
To reset the device to the default state:
`! U1 setvar "ip.port" "9100" (All printers except mobile)`
`! U1 setvar "ip.port" "6101" (Mobile printers)`
Return to Command List		

Note: Port numbers cannot be the same as any other SGD's in the group below. If you try to set the value to something that is in use it will be ignored. Setting the value to "0" disables the port and can be used to clear the current value, before it is set to a new value. However, remember setting the port to "0" will disable the port.

For further information on ports, refer to the Programming Guide.

For example:

```
ip.port
ip.port_alternate
ip.port_json_config
ip.port_single_conn
```

Note: Mobile printers use ip.port 6101 and ip.port_alternate is 9100.
Everything else uses ip.port 9100 and ip.port_alternate 6101.

Communications Commands

ALTERNATE TCP RAW PORT	Supported Printer Types	
Description: This is a secondary raw port that can be used to communicate with the printer.		

Considerations: Secondary raw printing port that allows multiple connections to the printer. These are served on and first come first served basis and allow up to x connection before additional connections are refused. This is primarily used for CPCL based printers and there to support legacy application. If ZPL is being used this port could be disabled without any impact. If this port is not being used, setting the value to 0 will disable the port.

Control Commands: The IP Port alternative capability is controlled by the ip.port_alternate command

To set the command:

```
! U1 setvar "ip.port_alternate" "6101"
! U1 setvar "ip.port_alternate" "0" (Disables port)
```

To confirm the command is set:

```
! U1 getvar "ip.port_alternate"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.port_alternate" "6101" (All printers except QLn)
! U1 setvar "ip.port_alternate" "9100" (QLn)
```

Return to Command List

Note: Port numbers cannot be the same as any other SGD's in the group below. If you try to set the value to something that is in use it, will be ignored. Setting the value to "0" disables the port and can be used to clear the current value, before it is set to a new value. However, remember setting the port to "0" will disable the port.

For further information on ports refer to the Programming Guide.

For example:

```
ip.port
ip.port_alternate
ip.port_json_config
ip.port_single_conn
```

Note: Mobile printers use ip.port 6101 and ip.port_alternate is 9100.
Everything else uses ip.port 9100 and ip.port_alternate 6101

Communications Commands

JSON RAW PORT	Supported Printer Types	
Description: This is a JSON port that can be used to send configuration commands to the printer.		

Considerations: This port is used to carry out printer configuration utilizing the JSON format and generally used by Zebra Applications and Utilities (PPME included), which would include 3rd party applications built using our SDKs. If this port is disabled, printers can still be recognized by PPME but communication will be slower.

Control Commands: The JSON port capability is controlled by the ip.port_json_config command

To set the command:

```
! U1 setvar "ip.port_json_config" "9200"
! U1 setvar "ip.port_json_config" "0" (Disables port)
```

To confirm the command is set:

```
! U1 getvar "ip.port_json_config"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.port_json_config" "9200"
```

Return to Command List

Note: Port numbers cannot be the same as any other SGD's in the group below. If you try to set the value to something that is in use it will be ignored. Setting the value to "0" disables the port and can be used to clear the current value, before it is set to a new value. However, remember setting the port to "0" will disable the port.

For further information on ports refer to the Programming Guide.

For example:

```
ip.port
ip.port_alternate
ip.port_json_config
ip.port_single_conn
```

Note: Mobile printers use ip.port 6101 and ip.port_alternate is 9100.
Everything else uses ip.port 9100 and ip.port_alternate 6101.

TCP RAW PORT (single)	Supported Printer Types	
Description: This port is used to send commands or files that the printer will act upon over a single TCP connection (this can include, CPCL, EPL, ZPL and Set-Get-Do commands). This is also known as the printer command language parser.		

Considerations: This port is designed to work in the same way as ip.port but it will only allow a single connection to the printer at a time. Any other connection attempts while this port is in use will be rejected.

Control Commands: The IP port single connection capability is controlled by the ip.port_single_conn command

To set the command:

```
    ! U1 setvar "ip.port_single_conn" "9300"
    ! U1 setvar "ip.port_single_conn" "0" (Disables port)
```

To confirm the command is set:

```
    ! U1 getvar "ip.port_single_conn"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "ip.port_single_conn" "9300"
```

Return to Command List

Note: Port numbers cannot be the same as any other SGD's in the group below. If you try to set the value to something that is in use it will be ignored. Setting the value to "0" disables the port and can be used to clear the current value, before it is set to a new value. However, remember setting the port to "0" will disable the port.

For further information on ports refer to the Programming Guide.

For example:

```
    ip.port
    ip.port_alternate
    ip.port_json_config
    ip.port_single_conn
    ip.port_single_conn_idle_timeout
```

Note: Mobile printers use ip.port is 6101 and ip.port_alternate is 9100.
Everything else uses ip.port 9100 and ip.port_alternate 6101

Communications Commands

TLS RAW PORT	Supported Printer Types	
Description: This port is used to send commands or files that the printer will act upon over a secure TLS channel (this can include, CPCL, EPL, ZPL and Set-Get-Do commands). This is also known as the printer command language parser.		

Considerations: This port is designed to work in the same way as ip.port but it requires a valid certificate loaded on the printer to enable TLS encryption. If you are using the TLS channel it is recommended that you disable the non-encrypted ports.

Control Commands: The TLS RAW Port connection capability is controlled by the ip.tls.port command

To set the command:

```
! U1 setvar "ip.tls.port" "9143"
! U1 setvar "ip.tls.port" "0" (Disables port)
```

To confirm the command is set:

```
! U1 getvar "ip.tls.port"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.tls.port" "9143"
```

Return to Command List

Note: This command requires that `ip.tls.enable` is on and that a valid certificate is present on the printer.

The certificate and private key can be deployed to the device as a single file, or separate files. If using a single file, the name of the file must be:

```
TLSRAW_CERT.NRD
```

If using multiple files:

```
TLSRAW_CERT.NRD – certificate file
TLSRAW_KEY.NRD – private key file
```

Once TLS communication is verified and operational, it is a security best practice to disable unencrypted forms of communicating with the printer over a network.

Communications Commands

TLS JSON PORT	Supported Printer Types	
Description: This is a TLS JSON port that can be used to send configuration commands to the printer over a secure connection.		

Considerations: This port is used to carry out printer configuration utilizing the JSON format and when utilizing the TLS connection.

Control Commands: The TLS connection JSON config port capability is controlled by the ip.tls.port_json_config command

To set the command:

```
! U1 setvar "ip.tls.port_json_config" "9243"
! U1 setvar "ip.tls.port_json_config" "0" (Disables port)
```

To confirm the command is set:

```
! U1 getvar "ip.tls.port_json_config"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.tls.port_json_config" "9243"
```

Return to Command List

Note: The value for TLS JSON Port may not be the same as another service already in use. If you try to set the value to something that is in use, it will be ignored. Setting the value to "0" effectively clears the current value and disables the port.

For further information on ports, refer to the Programming Guide.

For example:

```
ip.tls.port
ip.tls.port_json_config
```

Communications Commands

<table>
<tr><th>TLS ENABLE</th><th colspan="2">Supported Printer Types</th></tr>
<tr><td><b>Description:</b> This is a command that enables or disables the TLS capability.</td><td></td><td></td></tr>
<tr><td colspan="3"><b>Considerations:</b> This is for securing communications to the printer over wired and wireless Ethernet and depends on preloaded certificates on the printer. Ensure that this capability is working before disabling any non-TLS connections.</td></tr>
<tr><td colspan="3"><b>Control Commands:</b> The TLS Enable command is controlled by the ip.tls.enable command
To set the command:
<pre>   ! U1 setvar "ip.tls.enable" "on"</pre>To confirm the command is set:
<pre>   ! U1 getvar "ip.tls.enable"</pre>The printer responds with the current setting value, or "?" if not supported.
To reset the device to the default state:
<pre>   ! U1 setvar "ip.tls.enable" "on"</pre>Return to Command List</td></tr>
</table>

Note: This command enables TLS communication with the printer and requires a valid certificate is present on the printer.

Once TLS communication is verified and operational, it is a security best practice to disable unencrypted forms of communicating with the printer over a network.

Communications Commands

ASSET VISIBILITY AGENT	Supported Printer Types	
Description: This command turns the Asset Visibility agent off or on.		

Considerations: This feature can connect a networked Link-OS printer to Zebra’s Asset Visibility Service (AVS). The Asset Visibility Service is a Zebra-managed service offering that provides Zebra partners and customers ‘at-a-glance’ visibility to analytical insights about their device health, utilization, and performance.

Control Commands: The Asset Visibility capability is controlled by the weblink.zebra_connector.enable command

To set the command:

```
! U1 setvar "weblink.zebra_connector.enable" "on"
! U1 setvar "weblink.zebra_connector.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "weblink.zebra_connector.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "weblink.zebra_connector.enable" "on"
```

Return to Command List

Communications Commands

MQTT	**Supported Printer Types**	
Description: This command turns the MQTT capability off or on.		

Considerations: This feature can connect a networked Link-OS printer to an MQTT broker over TLS. MQTT like Web Sockets is a protocol that can be used to manage the printer. The printer supports two simultaneous connections (conn1 and conn2) that must have a server address configured before attempting any MQTT communication. The port used for each connection is specified as part of the connection

Control Commands: The MQTT capability is controlled by the mqtt.enable command

To set the command:

```
! U1 setvar "mqtt.enable" "on"
! U1 setvar "mqtt.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "mqtt.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "mqtt.enable" "on"
```

Return to Command List

Note: For further information on MQTT refer to the Programming Guide.

Applications Commands

CAPTURE PORT	Supported Printer Types	
Description: This command specifies the port that should be monitored for user data.		

Considerations: The capture channel command will collect user data from the specified port and store it in the capture.channel1.data.raw. To disable the capture channel the port should be set to "off"

Control Commands: The capture channel capability is controlled by the capture.channel1.port command

To set the command:

```
! U1 setvar "capture.channel1.port" "serial"
! U1 setvar "capture.channel1.port" "usb"
! U1 setvar "capture.channel1.port" "bt"
! U1 setvar "capture.channel1.port" "parallel"
! U1 setvar "capture.channel1.port" "off"
```

To confirm the command is set:

```
! U1 getvar "capture.channel1.port"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "capture.channel1.port" "off"
```

Return to Command List

Applications Commands

XML PRINTING	Supported Printer Types	
Description: This command enables or disables the XML parsing capability in the printer	Zebra Link™	

Considerations: The XML enable command is primarily used to allow the variable data for a stored format to be passed to the printer in an XML format. This is often used in the Oracle environment and if disabled will stop the printer from printing. The XML Data can be in two distinct formats, one for Oracle and one for SAP.

Control Commands: The XML capability is controlled by the device.xml.enable command

To set the command:

```
     ! U1 setvar "device.xml.enable" "on"
     ! U1 setvar "device.xml.enable" "off"
```

To confirm the command is set:

```
     ! U1 getvar "device.xml.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
     ! U1 setvar "device.xml.enable" "on"
```

Return to Command List

Applications Commands

USB MIRROR	Supported Printer Types	
Description: This command enables or disables the ability to perform mirroring using a USB device memory stick.		

Considerations: The USB mirror capability is only supported by printers that have USB host capability.

Control Commands: The USB mirror enabled capability is controlled by the usb.mirror.enable command

To set the command:

```
    ! U1 setvar "usb.mirror.enable" "on"
    ! U1 setvar "usb.mirror.enable" "off"
```

To confirm the command is set:

```
    ! U1 getvar "usb.mirror.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
    ! U1 setvar "usb.mirror.enable" "on"
```

Return to Command List

Note: This command only works on printers with USB Host capabilities.

Applications Commands

SYSLOG	Supported Printer Types	
Description: The printer can collect logging events and store them in non-volatile memory for analysis and debugging.		

Considerations: The syslog enable command turns on the logging capability which is turned off by default. There are other commands that configure the content of the file and max file size etc.

Control Commands: The syslog capability is controlled by the device.syslog.enable command

To set the command:

```
! U1 setvar "device.syslog.enable" "on"
! U1 setvar "device.syslog.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "device.syslog.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "device.syslog.enable" "off"
```

Return to Command List

Note: For further information on the syslog command refer to the Programming Guide.

For example:

```
device.syslog.clear_log
device.syslog.configuration
device.syslog.entries
device.syslog.log_max_file_size
device.syslog.save_local_file
```

Applications Commands

ZEBRA BASIC INTERPRETER	Supported Printer Types	
Description: This is to control the Zebra Basic Interpreter (ZBI) capability in the printer.		

Considerations: The ZBI enable command allows an administrator to enable/disable the ZBI Interpreter in the printer. A license is still required to be able to run ZBI scripts on a printer, however this is a global command to turn off the ZBI capability whether a license is installed or not. If you are not utilizing a ZBI script it is recommended that this is disabled.

Control Commands: The ZBI enable capability is controlled by the zbi.enable command

To set the command:

```
! U1 setvar "zbi.enable" "on"
! U1 setvar "zbi.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "zbi.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "zbi.enable" "on"
```

Return to Command List

Applications Commands

APL EMULATIONS	Supported Printer Types	
Description: This is to control the Advanced Printer Language (APL) Emulations capability in the printer.		

Considerations: The APL enable command allows an administrator to enable/disable the APL emulations in the printer. A valid file loaded on the printer is required to be able to run that emulation on a printer, however this is a global command to turn off the APL Emulation capability whether a file is loaded or not. If you are not utilizing APL Emulations, it is recommended that this is disabled.

Control Commands: The APL Emulations enable capability is controlled by the apl.enable command

To set the command:

```
! U1 setvar "apl.enable" "on"
! U1 setvar "apl.enable" "off"
```

To confirm the command is set:

```
! U1 getvar "apl.enable"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "apl.enable" "on"
```

Return to Command List

User Interface

PASSWORD	Supported Printer Types	
Description: This is the define password command and allows an admin to change the password for the Front Panel	Zebra Link™	

Considerations: The command allows the changing of the default password for Front Panel switches. The default password is well known and should be changed. It should also be noted that defaulting the password is trivial.

Control Commands: The Define Password capability is controlled by the ^KP command

To set the command:

```
^XA
^KPxxxx – where xxxx is any four-digit numeric sequence.
^JUS
^XZ
```

To confirm the command is set:

Use the Front Panel and attempt to modify a password protected configuration.

To reset the device to the default state:

```
^XA
^JUF
^XZ
```

Return to Command List

Note: The default password is "1234". Since it is documented and well-known default, it is recommended to change the password to something other than the default. It is also a good idea to change the Web Page password as it has the same default value. (See ip.http.admin_password command)

User Interface

ADMIN PASSWORD	Supported Printer Types	
Description: This is the define password command and allows the changing of the password for the web page	Zebra Link™	

Considerations: The command allows the changing of the default password for the web page access. The default password is well known and should be changed. It should also be noted that defaulting the password is trivial.

Control Commands: The password capability is controlled by the ip.http.admin_password command

To set the command:

```
     ! U1 setvar "ip.http.admin_password" "A%29921Hgg"
```

To confirm the command is set:

```
     ! U1 getvar "ip.http.admin_password"
```

The printer will only respond with a single "*" irrespective of the length of the password.

To reset the device to the default state:

```
     ! U1 setvar "ip.http.admin_password" "1234"
```

Return to Command List

Note: The default password is 1234. Since it is documented and well-known default, it is recommended to change the password to something other than the default. It is also a good idea to change the value of the Front Panel password as the default is the same. (see ^KP command)

Note: Regarding the "ip.http.admin_password" and "ip.http.admin_name" commands, the minimum length = 0, the maximum length = 25, and valid characters include any character that can be passed as a string.

User Interface

USERNAME	Supported Printer Types	
Description: This is the define username command and allows an admin to change the username for the web page	Zebra Link™	

Considerations: The command allows the changing of the default username for web page access.

Control Commands: The username capability is controlled by the ip.http.admin_name command

To set the command:

```
! U1 setvar "ip.http.admin_name" "Mainuser"
```

To confirm the command is set:

```
! U1 getvar "ip.http.admin_name"
```

The printer responds with the current setting value, or "?" if not supported.

To reset the device to the default state:

```
! U1 setvar "ip.http.admin_name" "admin"
```

Return to Command List

Note: The default username is "admin" and it can be changed, however there can only be one username.

Note: Regarding the "ip.http.admin_password" and "ip.http.admin_name" commands, the minimum length = 0, the maximum length = 25, and valid characters include any character that can be passed as a string.

Best Practices - Protected Mode

With Zebra printers there are several ways to configure the printer so that unused services are turned off, reducing the threat surface of the printer. Once the printer is securely provisioned and configured, it can be put into Protected Mode. This disables unauthorized changes and locks the current configuration down until an admin authorizes updates.

This is achieved through making use of a JSON formatted protect command. This command incorporates authentication information that must be validated, as well as an operation type specifying what the command does. First, here is how to set the password for a previously non-protected printer.

```
{}{
  "protect":{
    "authentication":{
    "username":"admin",
    "password":"",
    "type":"basic"
    },
    "operation":"setup",
    "setup":{
    "username":"admin",
    "password":"<new password here>"
    }
  }
}
```

In the example above there is a general protect command followed by an authentication section, operation type, and setup section. The password is initially an empty string because it has not been configured yet. Link-OS 6 supports the basic authentication type and a single admin.

To set the password, it is necessary to issue a setup operation command. Inside the setup section it is necessary to specify a password of at least 14 characters. Again, only the admin user is supported. As the password is sensitive information, it is highly recommended to configure this over a secure channel or segregated provisioning network.

If the command is successful, the response status code will be zero:

```
{}{"protect":{"status":0,"operation":"setup"}}
```

If the command is not successful, the response status code will be non-zero. Please see the JSON Commands Response Codes table for the meaning of non-zero response codes.

To verify if the printer is in protected mode or not check the return of the SGD command

```
device.protected_mode
```

If the printer is not in Protected Mode the command will return "off". If the printer is in Protected Mode, the command will return "on".

Although not recommended, it is possible to force protected mode off. In this scenario it is best practice to leave the admin password configured such that an adversary will be prevented from re-enabling protect mode or locking the printer out with an unknown password. This can be achieved by using a separate operation. For example:

```
{}{
  "protect":{
    "authentication":{
    "username":"admin",
    "password":"<current password here>",
    "type":"basic"
    },
    "operation":"configure-one",
    "configure-one":{
    "protected-mode-allowed":"no"
    }
  }
}
```

If the command is successful, it should return:

```
{"protect":{"status":0,"operation":"configure-one","protected-mode-
allowed":"no"}}
```

If the command is not successful, the response status code will be non-zero. Please see the JSON Commands Response Codes table for the meaning of non-zero response codes.

In order to turn Protected Mode back on, only set protected-mode-allowed to yes. Once in Protected Mode, protected settings can only be changed with a set operation in a protect command. For example:

```
{}{
  "protect":{
    "authentication":{
    "username":"admin",
    "password":"<current password here>",
    "type":"basic"
    },
    "operation":"set",
    "set":{
     "wlan.essid":"125",
    "usb.host.lock_out":"true"
    }
  }
}
```

If the command was successful, the printer will respond with:

```
{"protect":{"status":0,"operation":"set","set":{"wlan.essid":"125",
"usb.host.lock_out":"true"}}}
```

If the command is not successful, the response status code will be non-zero. Please see the JSON Commands Response Codes table for the meaning of non-zero response codes.

Not every setting on the printer is considered protected however, as there are many valid reasons to perform actions such as changing darkness between batches of print media. In general, settings related to network or security configuration are protected, whereas print settings are not. Any setting can be set within a valid protect set command whether it is protected or not. But once Protected Mode is enabled, protected settings can only be modified inside a protect command or until protected mode is disabled. To get the full list of protected settings issue the following command:

```
{}{"allconfig":null}
```

This will return all the settings the printer is capable of configuring and also includes an item for groups. If the groups value is set to a value of "1" it is protected. If it is "0" it is not protected and can be modified normally. Commands that are linked to other commands are NOT shown in the allconfig output. Please see Protected SGD Commands for more details.

To disable Protected Mode, re-enter the current password in the authentication section, do a setup operation, and in the setup section, use a password of empty string. Protected mode can also be disabled with a Decommission operation as described below.

Recommendation: Enable protected mode on the printer to prevent unwanted configuration changes. Any attempts to send unauthorized settings changes from any app or source are rejected when the printer is in Protected Mode.

Best Practices – Printer OS Download Protection

Zebra Link-OS printers use robust security mechanisms to ensure the authenticity and integrity of the printer OS download. Like Protected Mode, it is recommended that the ability to update the Printer's OS be restricted. To achieve this, Link-OS 6 has introduced a new SGD setting to prevent the firmware version from being changed. ("device.allow_firmware_downloads")

Recommendation:

Set the "device.allow_firmware_downloads" SGD to "no" and enable Protected Mode to ensure that the Download Protection setting cannot altered on unless an admin authorizes it

Just like other devices, printers require regular OS updates to stay current with functional and security fixes. It is best practice to establish a regular cadence of updating printers with the latest version. Upgrades work best when part of a planned process as it involves limited offline downtime to process the new firmware. When the time is right to upgrade a printer, the setting must be changed to allow new printer OS. This can be achieved in one of two ways.

One option is to bring the printer into a secure provisioning location, enable Printer OS downloads with an authorized protect command, download the update, and then disable Printer OS downloads again with a second authorized protect command. However, this involves a lot of steps and may be more complicated than necessary.

A second option is to utilize the Protected Mode operation "allow-next-firmware-download". This enables the printer to receive an authorized command from an admin to accept the next Printer OS download it receives while still powered on. After the update is processed the printer reverts back to not allowing any Printer OS to be downloaded.

Here is the command to perform this operation:

```
{}{
 "protect":{
  "authentication":{
   "username":"admin",
   "password":"<current password here>",
   "type":"basic"
  },
  "operation":"allow-next-firmware-download"
 }
}
```

If the command is successful, the printer will respond with the following response:

```
{} {"protect":{"status":0,"operation":"allow-next-firmware-download"}}
```

If the command is not successful it will respond with a non-zero "status" below are the possible values with their respective meaning.

Recommendation: Utilize the protected mode command to temporarily enable Printer OS downloads when an upgrade is desired

Best Practices - Certificates

A certificate consists of public information identifying the device and a set of public and private keys used for encrypted communication. This section discusses, in general, some best practice considerations for creating and using certificates for network services.

PKI Recommendations

PKI, or public key infrastructure, refers to the organization, creation, maintenance, and disposal of certificates in use for your devices. This section will not exhaustively detail all the best practices for PKI; it will touch on key points to consider for using certificates on your printer.

Certificate Files

The certificate and private key can be deployed to the device as a single file, or separate files. If using a single file, the name of the file must be:

```
XXXX_CERT.NRD
```

If using multiple files:

```
XXXX_CERT.NRD – certificate
XXXX_KEY.NRD – private key
```

The chain of trust file can support as many certificates as needed in this single file. For a PEM format, the two certificates would be concatenated together, one after the other.

The Chain of trust file is always loaded in:

```
XXXX_CA.NRD – certificate authority chain
```

Where XXXX is the name of the network service the certificates are intended for, acceptable values are:

```
WIRED
TLSRAW
HTTPS
MQTT1
MQTT2
WEBLINK1
WEBLINK2
```

WLAN certificates are an exception to this format and use different names:

```
CERTCLN.NRD - certificate
PRIVKEY.NRD - private key
CACERTSV.NRD - certificate authority chain
```

The printer supports PEM, and P12 certificate formats. It also supports DER files for WLAN and WIRED files.

Certificate Size Requirements

In keeping with latest industry wide recommendations (NIST, 2020), it is recommended to use only certificates with a signature of SHA-256 or higher (not including self-signed "Root" certificates).

For keys based on RSA, the size must be 2048 bits or higher. For keys based on ECDSA, the size must be 256 bits or higher.

Any certificates with a signature or key size smaller than this will not be accepted.

Unique Device Certificates

In general, a certificate is used to uniquely identify a device, determine ownership, and ensure you are communicating with the correct endpoint. The more times a single certificate is used on different devices, the more times the private key must be shared, which increases the risk that the information can be compromised. It is therefore recommended that each printer use its own unique certificate, preferably with a common name that contains the a unique identifier for the printer.

Certificate Life

The longer a certificate is in use, the higher chance it has of being compromised. It is therefore recommended to use the shortest valid certificate life as feasible with the printer in your network. A one-year expiration is the generally accepted recommendation for devices.

The printer has the capability of returning the expiration of the certificates it contains with an SGD command `file.cert.expiration`. This command will list all the network services that use certificates and any corresponding expiration information if a certificate is currently being used for that service. Here is an example of a printer that only contains the built-in certificates:

```
{"file.cert.expiration":[
{"service":"SHA1","file":"SHA1_DEVICE","expires_on":"2037-12-07 15:23:06"},
{"service":"SHA2","file":"SHA2_DEVICE","expires_on":"2028-11-11 09:56:05"},
{"service":"WLAN","file":null,"expires_on":null},
{"service":"WIRED","file":null,"expires_on":null},
{"service":"WEBLINK1","file":null,"expires_on":null},
{"service":"WEBLINK2","file":null,"expires_on":null},
{"service":"TLSRAW","file":null,"expires_on":null},
{"service":"HTTPS","file":null,"expires_on":null},
{"service":"MQTT1","file":null,"expires_on":null},
{"service":"MQTT2","file":null,"expires_on":null}]}
```

Certificate Creation

Off Printer, File Loaded

Certificates created on a separate system and loaded afterwards onto the printer file system has been supported since Link-OS 5 using the file naming scheme described earlier. This allows for support of existing workflows and older printer firmware. Unfortunately, it also involves transferring the private key from that system to the printer which increases the likelihood of the key material becoming compromised if not done securely.

Because certificates rely on sufficiently random numbers, you will want to ensure the system it is generated on has entropy sufficiently high for the creation of a new certificate and key. On Linux-based systems, this can be achieved by:

```
cat/proc/sys/kernel/random/entropy_avail
```

You will need to create certificates that contain the host name that the printer will have on the network as its common name in the certificate. As an example, here are some OpenSSL commands to achieve this:

RSA

```
openssl genrsa 2048 > XXXX_KEY.NRD
openssl req -new -x509 -nodes -sha256 -days 365 -key XXXX_KEY.NRD >
XXXX_CERT.NRD
```

You must fill out a valid Country, State, City, Company, and Common name.

ECC

```
openssl ecparam -out ec_params.pem -name prime256v1
openssl req -new -x509 -nodes -sha256 -days 365 -newkey ec:ec_params.pem -
keyout XXXX_KEY.NRD > XXXX_CERT.NRD
```

On Printer, CSR Generation

Generate CSR

A multipart form (MPF) command format is used to pass in parameters required for the printer to generate a new private key and a CSR file in PEM format. The printer automatically ensures enough entropy is present before attempting to generate the files. An alert is generated and sent over the weblink main connection or configured channel(s) when the CSR is ready. The CSR file can then be removed from the printer and sent to your signing authority where a certificate is created. That certificate can then be returned to the printer using the "place_cert" MPF command where it is validated before saving to the E drive.

Here is an example of the command:

```
{}--<boundary characters><CR><LF>
Content-Disposition: form-data; name="files"; filename="<service name>";
action="generate_csr"<CR><LF>
Content-Type: application/octet-stream<CR><LF>
Content-Transfer-Encoding: binary<CR><LF>
<CR><LF>
{"CN":"CL01",
  "key":{
    "algo":"ecdsa","size":256,"curve":"prime256v1"},
  "names":[
    {"C":"US",
    "L":"Lincolnshire",
    "O":"Zebra Technologies",
    "OU":"AIT",
    "ST":"Illinois",
    "challengePassword":"<challenge password>",
    "emailAddress":"<email address>",
    "subjectAltName":"<alternate name>"}],
  "message_digest":"sha256"}
<CR><LF>--<boundary characters>--
```

Where:

`{}` = Zebra defined starting characters used to signal the JSON parsing request

`--<boundary characters>` = must start with `--` and must contain no control characters (less than 0x20) until it ends with an end of line which is `<CR><LF>`. This is a group of characters that the exact sequence is not contained in the rest of the request. The boundary must be no more than 72 character which includes the `--`.

`<CR><LF>` = each line ends with a carriage return and line feed characters (0x0A 0x0D*)*

<`service name`> is the name of the service for which you want a CSR to be generated. It is case sensitive. Non-supported values will cause an error response. Acceptable values are:

```
WLAN
WIRED
TLSRAW
HTTPS
MQTT1
MQTT2
WEBLINK1
WEBLINK2
```

Following the headers and an additional `<CR><LF>` the body data for the CSR request begins

"`CN`" is the common name for the certificate

"`key`" requires "`algo`" and either "`size`" or "`curve`" field

  "`algo`" is the algorithm field. Supported values are "`rsa`" and "`ecdsa`"

  "`size`" is the key size.

    "`rsa`" supports `2048`, `3096`, and `4098`

    "`ecdsa`" if "curve" is missing, can be the following (and the implied curve):

      `224` (secp224r1)

      `256` (secp256r1)

      `384` (secp384r1)

      `521` (secp521r1)

  "`curve`" is the name of the curve.

    Use "`file.cert.curves`" to get a list of supported curves

"`names`" contains fields put into the CSR request and may include

  "`C`" Country

  "`L`" Locality

  "`O`" Organization

"ST" State
"OU" Organizational Unit
"emailAddress" Email Address
"subjectAltName" Subject Alternative Name (SAN)

Without a format specifier the SAN will be considered a DNS name by default

"subjectAltName":"website.com" --> This will produce a CSR with a DNS SAN entry of "website.com"

With an OpenSSL format specifier, other types of SAN can be used such as User Principal Name (UPN), email address, IP address, and DNS.

"subjectAltName":"otherName:msUPN;UTF8:a@b.org" --> This will produce a CSR with a UPN SAN of "a@b.org"

Can be a single JSON string as above, or a JSON array of strings

"subjectAltName":["website.com", "otherName:msUPN;UTF8:a@b.org"] --> This will produce a CSR with a SAN containing both of the previous SAN types

Please reference OpenSSL documentation for more details

"challengePassword" Challenge Password required for some PKI systems

"message_digest" can be: sha256, sha384 or sha512

<CR><LF>--<boundary character> = Terminates a multipart request. If you have another request you may start with the next headers immediately, no additional boundary needed. When you have no more requests add an additional – characters (2 dashes) to terminate the multipart form parsing. You mix and match any combination of multipart form requests.

Successful generation will cause a CSR to be generated on the printer named

```
CSR_<SERVICE_NAME>_CERT.CSR
```

The CSR will be placed in protected space on the Z drive while the private key generated will be placed on the E drive. You can retrieve this file from the printer via MPF "retrieve" command or other means.

Return response and alert

The generate_csr command always returns a response. It will return status="processing" if it has found no errors in the request.

```
[{"action"="generate_csr","filename":"<filename>","status"="processing"}]
```

It will return status="error" error_code=<number> if it has encountered an error in the request. If there is an error in the request it will not try to generate a CSR file.

```
[{"action"="generate_csr","filename":"<filename>","status"="error",
"error_code":42}]
```

Here are some error codes that can be returned:

```
No error = 0
System error = 1
Bad service name string, pick one of the supported services = 10
A problem occurred with the CRC when placing a cert over MPF = 11
A problem occurred with the file size when placing a cert over MPF = 12
Too many CSR requests are being processed = 50
Bad JSON formatting = 51
Bad common name (CN) = 53
Bad key size for the algorithm, pick a supported size = 54
Bad curve name, pick a supported name = 55
Bad key algorithm type, pick a supported type = 56
Bad location = 57
Bad state = 58
Bad country = 59
Bad organization = 60
Bad organization unit = 61
Bad email = 62
Bad subject alternate name = 63
Bad digest type, pick a supported digest type = 64
The challenge password specified was too large = 65
The challenge specified was invalid = 67
The customer organization unit was too large = 68
The hardware common name was invalid = 69
```

Once the request is processing an alert will be returned over the weblink main connection or configured channels when the certificate processing is complete. It may be successful or an error. The alert generated looks like this:

```
{
 "alert" : {
 "unique_id" : "XXXYYYZZZ",
 "time_stamp" : "2015-06-09 03:38:12",
 "type_id" : "ALERT or ERROR",
 "condition_id" : "CSR AVAILABLE",
 "condition_state" : "SET",
 "type" : "ALERT or ERROR CONDITION",
 "condition" : "CSR AVAILABLE ",
 "filename" : "UserCert.csr",
 "condition_code" : 0
 }
}
```

Where:

`unique_id` Printer Serial Number, as it appears on printer label
`time_stamp` Date/Time when the alert is generated
`type_id` "`ERROR`" if CSR generation failed or "`ALERT`" if success
`condition_id` Always "`CSR AVAILABLE`", identifies the alert
`condition_state` Always "`SET`"
`type`
"`ERROR CONDITION`" if CSR generation failed
"`ALERT`" if success
`condition` Always "`CSR AVAILABLE`"
`filename` The filename of the generated CSR (extension always .csr)
`condition_code` error code, listed above

Supported ECDSA curves

The following setting can be used to determine the available ECDSA curves that the printer supports:

```
{"file.cert.supported_curves":"secp224r1,secp256r1,secp384r1,secp521r1,bp256r1,bp3
84r1,bp512r1,curve25519,secp224k1,secp256k1,curve448"}
```

Place Cert

A multipart form format to place a certificate onto the printer for usage by the printer. It will try to pair the public key in the certificate with the previously generated private key on the printer. If the private key is not found or there is a mismatch, an error will be returned.

```
{}--<boundary characters><CR><LF>
Content-Disposition: form-data; name="files"; filename="<service name>";
action="place_cert"<CR><LF>
Content-Type: application/octet-stream<CR><LF>
Content-Transfer-Encoding: binary<CR><LF>
<CR><LF>
<pem cert data>
<CR><LF>--<boundary characters>--
```

Where:

`{}` = (defined in generate CSR section)

`--<boundary characters>` = (defined in generate CSR section)

`<CR><LF>` = (defined in generate CSR section)

<`service name`> is the name of the service for which you want a to place the signed certificate. It is case sensitive. Other values will cause and error response. Successful placement will cause the private key and certificate to be placed into usage for that service. This should match the same service name used when generating the CSR

<`pem cert data`> is the actual PEM file contents of the signed certificate

`<CR><LF>--<boundary character>` = (defined in generate CSR section)

The place_cert command always returns a response. It will return status="`success`" if it has received a valid certificate, found the matching private key, and place the files into service.

[{"action"="place_cert",filename":"<filename>","status"="success"}]

It will return status="`error`" `error_code`=<number> if it has encountered an error in the request.

[{"action"="place_cert",filename":"<filename>","status"="error","error_code":42}]

MPF response error codes:

```
Certificate being placed is formatted incorrectly = 150
Certificate being placed using a weak cipher = 151
Certificate being placed does not match private key generated = 152
Certificate being placed is not valid for time on printer = 153
```

Supported Ciphers

The following ciphers are supported for Weblink, HTTPS, and TLS. When setting up your system to communicate, you should use a secure cipher to help prevent the connection from being compromised. We would suggest at least `DH-RSA-AES128-SHA256` but the following are all supported by Link-OS 6:

```
ECDHE-ECDSA-AES256-GCM-SHA384
ECDH-RSA-AES256-GCM-SHA384
ECDH-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES128-GCM-SHA256
ECDH-RSA-AES128-GCM-SHA256
ECDH-ECDSA-AES128-GCM-SHA256
DH-DSS-AES256-GCM-SHA384
DH-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-GCM-SHA384
DH-DSS-AES128-GCM-SHA256
DH-RSA-AES128-GCM-SHA256
DHE-RSA-AES128-GCM-SHA256
ECDHE-RSA-AES256-SHA384
ECDHE-ECDSA-AES256-SHA384
ECDH-RSA-AES256-SHA384
ECDH-ECDSA-AES256-SHA384
DHE-RSA-AES256-SHA256
DH-RSA-AES256-SHA256
DH-DSS-AES256-SHA256
ECDHE-RSA-AES128-SHA256
ECDHE-ECDSA-AES128-SHA256
ECDH-RSA-AES128-SHA256
ECDH-ECDSA-AES128-SHA256
DHE-RSA-AES128-SHA256
DH-RSA-AES128-SHA256
DH-DSS-AES128-SHA256
AES256-GCM-SHA384
AES128-GCM-SHA256
AES256-SHA256
AES128-SHA256
```

Certificate Downloading

Certificates themselves do not contain any data that must be kept private. A private key on the other hand must be kept secure to prevent being exposed. It is security best practice to load certificates to the printer in a secure provisioning environment over an encrypted channel such as TLS. Secure provisioning networks are typically segregated from production or widely available networks. If encryption is unavailable, a physical connection such as USB is recommended. To download the various certificate files to the printer, choose one of the following methods in security preferred order:

Note: Use the appropriate file name as discussed in the Certificates Best Practices section of this document.

1. **Multipart Form Store: See the multipart form section elsewhere in this document**
2. **SDK:** Use the Zebra Multiplatform SDK command line STORE function to send the files to the printer. The SDK is available for download at www.zebra.com/sdk

3. **ZPL:** Use the ! CISDSFCRC16 command, with the appropriate headers to the certificate to store the files on E: drive of the printer. Details available in the ZPL Programming Guide, available at www.zebra.com.

   Use the ~DY command, with the appropriate header to the certificate to store the files on E: drive of the printer. Details available in the ZPL Programming Guide, available at www.zebra.com.

4. **FTP:** If using FTP, make sure that the printer’s "execute file" function is turned off while you send the file, so the file is stored and not processed as a printing command. This can be done by sending the following command:

```
! U1 setvar "ip.ftp.execute_file" "off"
```

Note: The command must be followed by a carriage return or a space character. If you plan on using FTP for printing purposes, be sure to reset this feature to "on" after storing the certificate files.

Connect to the printer via FTP and download the certificates to the printer.

Validating Certificates

To validate that your certificates are loaded onto the printer correctly, choose one of the following methods.

1. JSON: Issue the following to get a list of files on E drive. Those downloaded via Multipart form will also list the CRC32 such that you can assure that the file you have matches the file on the printer.

```
{}{"file.drive_listing":"E"}
```

2. ZPL: Issuing one of the following commands allows you to confirm that the certificates have been stored on the file system. This can be done utilizing a terminal program or Zebra Setup Utilities.

```
^XA^WDE:*.nrd^XZ
```

Note: The above command will print a label listing all the files on the E: drive that have the ".nrd" extension.

```
^XA^HWE:*.NRD^XZ
```

Note: The above command will transmit a listing back to the host with all the files on the E: drive that have the ".nrd" extension.

3. Internal Web Page: Log into the internal web page and select Directory Listing.

You will be able to confirm that the certificate files are on the file system. However, you will only be able to see the files; you not be able to download them or view the contents.

Deleting Certificates

To delete certificates loaded on the printer, use the following method.

1. JSON: Issue the following command over any connection to delete the file you specify in place of CERTNAME.NRD.

```
{}{"file.delete":"E:CERTNAME.NRD"}
```

2. ZPL:

   a. Issuing the following command allows you to delete a certificate file stored on the file system. This can be done utilizing a terminal program or Zebra Setup Utilities.

```
^XA^IDE:<CERTNAME>.NRD^XZ
```

where `<CERTNAME>` is a single certificate file name.

or

```
^XA^IDE:*.NRD^XZ
```

This will delete all files with the .nrd extension.

   b. Issuing the following SGD command allows you to delete the specified file stored on the file system.

```
! U1 do "file.delete" "value"
```

Best Practices - WLAN Certificates

As described in the certificate best practices section it is important to use unique certificates per device to minimize access to the private key. Both can be achieved using the printer CSR (certificate signing request) functionality.

Starting in Link-OS 6 the printer supports JSON multi-part form commands for generating CSRs as well as placing the CA signed certificate back onto the printer. There is support for different message digests, ciphers, and key lengths to best meet a variety of security needs.

Private Key Passphrase

The client private key can be optionally encrypted with a passphrase. This is useful if the private key file is in an unprotected part of your network or needs to be transmitted in the clear.

It is important to note that the passphrase itself is not stored in an encrypted fashion on the printer. Because the passphrase must be kept secure, it is a best practice to configure this passphrase over a physical connection (USB), or a segregated provisioning network that is separate from the production or company network. The private key passphrase can be configured with the following SGD:

```
wlan.private_key_password
```

Certificate Files

```
CERTCLN.NRD - certificate
PRIVKEY.NRD - private key (optionally encrypted)
CACERTSV.NRD - certificate authority chain
```

For the certificate authority chain, if one access point certificate was signed by one CA, and another access point certificate was signed by a different CA, the same trust file could be used for both APs as long as both signing certificates were included in the same trust file.

Automation

It is recommended that you automate the process of renewing WLAN certificates. Printer Profile Manager Enterprise (PPME) version 3.1 or later can automate this process for you. Outlined below is the process PPME uses in certificate renewal process:

1. Poll the printer for certificate expiration date and time, on an interval dependent on your certificate lifetime
2. Determine if the WLAN certificate should be renewed or not
3. If the certificate should be renewed, issue a generate_csr command to the printer
4. Once ready, retrieve the CSR from the printer
5. Sign the CSR with a CA
6. Use the "place_cert command" to put that signed certificate back on the printer
7. Plan a time to reset the printer so that the new certificate can be used

If the printer already contains a CSR it can be reused by the CA and signed again without the printer needing to recreate the CSR. This assumes the private key has not been compromised.

Best Practices - LAN 802.1x

802.1x over LAN provides a mechanism to authenticate devices connecting to a network. To get this set up on the printer, a few settings must be configured. Once configured, the settings will take effect after a reset.

Security

The printer currently supports `peap`, `eap-tls`, and `eap-ttls` security. The choice of printer authentication mode should be driven by what is already in place on your network. In general, `eap-tls` provides a more robust mutual authentication and requires client certificates. If starting from scratch and with a robust PKI (public key infrastructure) already in place, `eap-tls` provides a more secure option, but may be more challenging to deploy. You can select your security method by using the following SGD command:

```
internal_wired.8021x.security
```

Username

The username is something that is needed for connection to the network and can be configured with the following SGD:

```
internal_wired.8021x.username
```

Private Key Passphrase

The client private key for use with TLS security can be optionally encrypted with a passphrase. This is useful if the private key file is in an unprotected part of your network or needs to be transmitted in the clear.

It is important to note that the passphrase itself is not stored in an encrypted fashion on the printer. Because the passphrase must be kept secure, it is a best practice to configure this passphrase over a physical connection (USB), or a segregated provisioning network that is separate from the production or company network. The private key passphrase can be configured with the following SGD:

```
internal_wired.8021x.private_key_password
```

Certificate Files

The certificate filename prefix is WIRED

```
WIRED_CERT.NRD – certificate file
WIRED_KEY.NRD – private key file (optionally encrypted)
WIRED_CA.NRD – certificate authority file
```

The Certificate authority file is `for the certificate received from the RADIUS server. This is used by the printer to verify the server's identity.` The printer supports PEM, DER, and P12 certificate formats.

Best Practices - Bluetooth Security

Bluetooth security on Link-OS printers is very important when deploying large numbers of remotely accessible devices into a customer site. Many times, Bluetooth-enabled Zebra devices will follow associates for the duration of a shift - and come into range of the public many times during that shift.

The goal of securing Bluetooth-enabled Zebra printers is to prevent unauthorized access to the printer from a distance. Certain information and profiles can be accessed by any remote device, but some profiles contain sensitive data and/or allow administrative capabilities. For these reasons, it is important to secure Bluetooth connected devices.

Overview

Transports

Bluetooth functionality is divided into two supported *transports*: Classic (also known as BR/EDR) and Low Energy (also known as BTLE or LE). Each transport has slightly different security features and considerations; this document will address them separately.

Some Bluetooth-capable Zebra printers support only Bluetooth Classic, some support only Bluetooth LE, and some support both.

Pairing and Encryption

Pairing in Bluetooth refers to a process in which you can associate two Bluetooth devices with a shared, private encryption key. The storage of these encryption keys for later use is referred to as *bonding.* It is important to note that once two Bluetooth devices are bonded, they are considered **trusted.** That is, future connections between those two devices will resume the encrypted session silently, and the remote device will retain access to sensitive profiles. This makes it crucial that two untrusted devices are never paired.

Authentication

Establishing an encrypted connection between two Bluetooth devices is not the only consideration for secure communications; it is often important to establish an *authenticated* connection in addition to an *encrypted* connection. An encrypted connection is considered authenticated if it can be proven that the connected devices exchanged encryption keys without a Man-in-the-Middle (MITM) being able to intercept the keys. Bluetooth uses distinct security procedures depending on whether devices can provide authenticated connections; these will be discussed below for both Classic and LE.

Bluetooth Classic

Discoverability

The SGD command "`bluetooth.discoverable`" controls whether the Zebra printer will respond to *inquiry requests* from a remote device. This Classic feature is called *discoverable mode*: if it is disabled, remote devices are not able to easily find the printer.

NOTE: Starting with Link-OS 6, the "`bluetooth.discoverable`" function is now **off** by default and other devices cannot see or connect to the printer.

With discoverability disabled, the printer will still make connections with a remote device that was previously paired.

RECOMMENDATION: Only keep discoverable mode enabled while paring to a remote device. Once paired, discoverable mode should be disabled. Starting with Link-OS 6, a new feature was introduced to enable limited discovery. Holding down the FEED button for 5 seconds will enable limited pairing mode. Limited pairing mode enables discovery and pairing for 2 minutes. This enables the printer to operate safely with discoverable mode disabled until a user with physical access to the printer activates it.

Upon entering Bluetooth Pairing Mode, the printer will provide feedback that the printer is in Pairing Mode using one of these methods:

- On printers with a "Bluetooth" screen icon or Bluetooth LED, the printer shall flash the "Bluetooth" screen icon or Bluetooth LED on and off every second while in pairing mode
- On printers without a "Bluetooth" screen icon or Bluetooth LED, the printer shall flash the "Data" icon or Data LED on and off every second while in pairing mode
- Specifically, on the ZD220, ZD230, and ZD888 models, the 4 flash LED sequence places the printer into Bluetooth Pairing Mode.
- Specifically, on the ZD510 model, the 5 flash LED sequence places the printer into Bluetooth Pairing Mode.

NOTE: If the user wants to completely disable Bluetooth connectivity, including discovery and pairing, they can disable the Bluetooth radio entirely.

Pairing

Bluetooth Classic security and pairing modes have evolved with revisions to the standard, and can be divided into three major groups:

1) **No security** – Neither encryption nor authentication are required to access sensitive profiles.

2) **Legacy security (pre-SSP)** – Prior to Bluetooth 2.1, Classic connections could only be secured with a "PIN"; this is a variable-length shared passphrase that allows two devices to start encryption and pairing. Any sequence of bytes may be used to form a PIN, including ASCII characters. It is not limited to numeric values, although not all Bluetooth devices support alphanumeric PIN entry.

3) **Secure Simple Pairing (SSP)** – With the introduction of Bluetooth 2.1, Secure Simple Pairing allows for several types of simple modes to encrypt and authenticate communications between two SSP-enabled devices. The modes available depend on the *I/O capabilities* of the two devices wishing to communicate and provide varying levels of authenticity guarantees and protection against MITM attacks.

When a device supporting SSP tries to access one of the printer’s Serial Port Profiles, SSP pairing will always be used. If both devices have a display and support MITM protection, the *Numeric Comparison* pairing procedure will be used. This procedure requires both sides to display and confirm a 6-digit numeric code that is securely exchanged between the two devices. If a third device attempts to Man-In-The-Middle the desired Bluetooth devices, the target devices will display different numeric codes and pairing should be rejected by the user.

If one or both devices do not support a display, the *Just Works* pairing procedure will be used, if allowed by the printer’s configuration. *Just* Works mode encrypts the connection, but no prompts will be shown by either side to confirm this process. There is no way to verify that a third device has not performed an MITM attack; *Just Works* is an unauthenticated pairing procedure.

Zebra printers also support "no security" and legacy PIN pairing modes to be backwards compatible with early Bluetooth radios and stacks, many of which are still in use by our customers. This feature is enabled by default. However, it is recommended that customers who do not need these modes disable them to prevent unauthorized access.

Bluetooth Classic security capabilities are controlled by four SGDs:

1. "`bluetooth.minimum_security_mode`": Selects minimum level of security required for a remote device to access all profiles and services on the printer.

    - "1" - No security is required.
    - "2" - Encryption is required; MITM protection is *not* required.
    - "3" - Encryption and MITM protection are required; legacy pairing is enabled.
    - "4" - Encryption and MITM protection are required; SSP is required. This will force Numeric Comparison mode.

2. "`bluetooth.allow_non_display_numeric_comparison`": for printers without a display, this setting controls whether the Numeric Comparison confirmation code is displayed by physically printing it (**default**), automatically confirming it, or disabling Numeric Comparison entirely.

3. "`bluetooth.bonding`": enable (**default**) or disable storage of link keys for paired printers. It is **not recommended** to disable this feature.

4. "`bluetooth.bluetooth_pin`": Configure the legacy PIN shared secret; the printer supports PINs up to the maximum of 16 bytes. If the PIN is empty, legacy PIN pairing is disabled. The PIN is **empty by default**.

NOTE: For printers that have a display, the minimum-security level default changed from 1 to 3 in Link-OS 6.

RECOMMENDATIONS: The recommended Bluetooth security configuration will depend on the types of printers in use and the remote devices connecting to them. If the remote devices expected to connect to Zebra printers have a display and support Secure Simple Pairing, and the Zebra printer has a display, it is highly recommended to configure the minimum security level to 4. This forces the remote device to use a pairing mode that supports MITM protection and will not allow legacy nor unencrypted access.

If the printer is a model without a display, it is a bit trickier to use minimum security level 4, as the numeric comparison code for SSP cannot be displayed. Such printers are configured by default to print the comparison code on the customer's media; however, this may not be desirable if frequent pairing is required or if the customer's media is expensive.

If the remote device does not support Bluetooth 2.1 with SSP, the minimum security level should be set to 3 and "`bluetooth.bluetooth_pin`" must be set to the desired shared secret. This forces MITM protection while allowing legacy PIN pairing. **Legacy PIN pairing is not recommended for new integrations.**

Bluetooth Low Energy (BTLE)

Advertising

The concept of *advertising* mode is similar to discoverable mode in Bluetooth Classic, with a few key differences. Unlike in Bluetooth Classic, Bluetooth LE devices are only connectable while they are advertising.

NOTE: Zebra printers do not currently support a capability to disable LE advertising without completely disabling Bluetooth LE support, which implies LE-enabled printers are always connectable. To disable Bluetooth LE on dual-mode (Classic+LE) printers, you can set the SGD `bluetooth.le.controller_mode` to "classic".

Pairing

Pairing in Bluetooth LE is similar to Classic; pairing can be both authenticated (with MITM protection) and unauthenticated. The SGD "`bluetooth.minimum_security_mode`" controls whether pairing/encryption is required to access the Zebra Printer and Configuration Service.

1. "`bluetooth.minimum_security_mode`": Selects minimum level of security required for a remote device to access all profiles and services on the printer.

   - "1" - No security is required.
   - "2" - Encryption is required; MITM protection is *not* required.
   - "3" or "4" - Encryption and MITM protection are required.
2. "`bluetooth.allow_non_display_numeric_comparison`" allows printers without a display to print the passkey or numeric comparison code on the user's media.

Much like Classic, LE supports a "Just Works" mode (no MITM protection) for devices without a display, and a "passkey" mode that is similar to "Numeric Comparison" on Classic.

LE versions 4.2+ also support a "Numeric Comparison" pairing mode; this is supported on printers with 4.2-compatible Bluetooth radios, and firmware versions Link-OS 5 and newer. Passkey and Numeric Comparison pairing modes provide MITM protection.

RECOMMENDATION: Force pairing requiring MITM support by setting "`bluetooth.minimum_security_mode`" to "4". If the printer cannot support display of the passkey, set it to "2".

Best Practices - HTTPS Security

Certificate Files

Starting in Link-OS 5, you can also communicate using HTTPS to view printer web pages over a TLS channel to ensure that communication is encrypted. To begin communicating with the printer over HTTPS, you first need to deploy a certificate to the device. A certificate consists of public information identifying the device and a set of public and private keys used for encrypted communication to the device.

Please note that any common name will be accepted by most browsers. However, you should select a common name that preferably contains the printer's host name.

The certificate filename prefix is HTTPS.

```
HTTPS_CERT.NRD – certificate file
HTTPS_KEY.NRD – private key file (cannot be encrypted)
HTTPS_CA.NRD – certificate authority chain
```

The certificate authority chain will be presented during connection to the client. It should contain all the appropriate intermediary certificates in the trust chain between the printer's certificate and a trusted authority.

HTTPS Port

Once the device certificates are loaded and the printer has rebooted, you can begin using HTTPS. The port for HTTPS is, by default 443, and can be configured using the following SGD command:

```
"ip.https.port"
```

This assumes that HTTPS is enabled with the following SGD command:

```
"ip.https.enable"
```

Disable HTTP Access

Once HTTPS communication is verified and operational, it is a security best practice to disable unencrypted forms of communicating with the printer over a network. This includes disabling HTTP access using the "`ip.http.enable`" command.

Public Key Validation

As stated earlier, the HTTPS implementation does no authentication of devices connecting to it. The client connecting to the printer can validate it is talking directly to the printer through the use of comparing public keys. The client should know the public key of the printer that was originally loaded. When making the first connection to the printer, the client can verify this pinned public key to the one it is currently receiving from the printer to ensure there is no Man In The Middle (MITM) interference occurring.

Best Practice - TLS Security

Disable Unsecure Network Access

Once TLS communication is verified and operational, it is a security best practice to disable unencrypted forms of communicating with the printer over a network. This includes disabling:

```
ip.tcp.enable
ip.udp.enable
ip.ftp.enable
ip.lpd.enable
ip.http.enable
ip.snmp.enable
```

Enable Firewall Allow list

It is important to note that in the steps above, we have only established encrypted communication, but not authentication. The printer accepts any connection over TLS and does no authentication of the host. As such, you could also ensure that only communication from the desired host IP address is allowed through use of the following SGD:

```
ip.firewall.whitelist_in
```

Public Key Validation

As stated earlier, the TLS implementation does no authentication of devices connecting to it. The client connecting to the printer can, however, validate it is, in fact, talking directly to the printer by comparing public keys. The client should know the public key of the printer that was originally loaded. When making first connection to the printer, the client can verify this pinned public key to the one it is currently receiving from the printer to ensure there is no Man In The Middle (MITM) interference occurring.

Best Practices - TCP Channel Security

TCP Configuration

TCP Raw Ports

Communication with the printer command language parser is enabled over TCP via multiple ports. For unencrypted TCP raw access, there are two ports available, 6101 and 9100, and may be configured respectively using the following SGD commands:

```
ip.port
ip.port_alternate
```

To make use of TCP raw communication, ensure that it is enabled using the following SGD command:

```
ip.tcp.enable
```

JSON Raw Port

In addition to the printer command language parser, JSON is used exclusively for configuration retrieval and modification with no label formatting support. This is accessible via a separate port, 9200, which is configurable using the following SGD command:

```
ip.port_json_config
```

TCP Raw Communication

To easily verify the printer is responding, you can connect to the printer via a telnet application using one of the ports specified above. Then, send a simple command to the printer (such as ~HI if it supports ZPL) to verify it was received and sends data back. You will also be able to view traffic unencrypted via any packet capturing software.

TLS Configuration

Certificate Files

Starting in Link-OS 5, you can also communicate using TLS to provide an encrypted channel to the printer. To begin communicating with the printer over TLS, you first need to deploy a certificate to the device. A certificate consists of public information identifying the device and a set of public and private keys used for encrypted communication to the device.

Please note that any common name will be accepted. However, you should select a common name that preferably contains the printer's host name.

The certificate filename prefix is TLSRAW

```
TLSRAW_CERT.NRD – certificate file
TLSRAW_KEY.NRD – private key file (cannot be encrypted)
TLSRAW_CA.NRD – certificate authority chain
```

The certificate authority chain will be presented during connection to the client. It should contain all the appropriate intermediary certificates in the trust chain between the printer's certificate and a trusted authority.

TLS Raw Port

Once the device certificates are loaded and the printer has rebooted, you can begin encrypted communication using TLS. The port for TLS connecting to the printer command language parser is, by default, 9143, and can be configured using the following SGD command:

```
ip.tls.port
```

This, of course, assumes that TLS is enabled using the following SGD command:

```
ip.tls.enable
```

TLS JSON Raw Port

As before, the printer also has a JSON interface for encrypted communication with TLS using port 9243, and can be configured using the following SGD command:

```
ip.tls.port_json_config
```

TLS Communication

To verify the printer is working with the device certificates over TLS, you can issue the following OpenSSL command:

```
echo "~WC" | openssl s_client – connect 10.80.124.159:9143 -quiet
```

This sends the ~WC ZPL print config label command to openssl for a TLS connection to the printer and port specified. If you attempt to view captured packets, you will also find that the data is encrypted and unreadable.

Best Practices - Weblink (Web Sockets) Security

Certificates

By default, the printer comes supplied with a generic weblink device certificate and Zebra server certificate authority. These certificates can be used for connecting to a weblink or web sockets server with a Zebra signed server certificate.

Another option is to use Link-OS 5 or greater user supplied certificates. Individual certs are best but general printer certificates can be used with care as well. Upon reset, once the printer has an IP address, it will attempt to use the provided certificates to make an initial weblink connection.

Certificate Files

The following filenames shall be used to store the certificates:

```
WEBLINKX_CERT.NRD – device printer certificate
WEBLINKX_KEY.NRD – device printer private key (cannot be encrypted)
WEBLINKX_CA.NRD – server certificate authority chain
WEBLINKX_CRL.NRD – certificate revocation list
```

Where "WEBLINKX" is either "WEBLINK1" or "WEBLINK2"

Retry Interval

To prevent flooding a weblink server with connections, it is recommended to configure a random retry interval. This allows for all the devices connecting to the weblink server to attempt reconnection at different times after a connection loss event. The SGD to configure this is:

```
weblink.ip.connX.retry_interval_random_max
```

Where `connX` is the connection 1 or 2 for weblink

If this is set to a non-zero value, the printer will wait a random number of seconds between 1 and the value specified when attempting to reconnect. If the value is zero, then another SGD will be used to configure the number of seconds it will wait before attempting reconnection. The SGD to configure this is:

```
weblink.ip.connX.retry_interval
```

Where `connX` is the connection 1 or 2 for weblink

How to Create a Weblink Server CSR (certificate signing request)

For detailed steps please visit: https://techdocs.zebra.com/link-os/2-14/webservices/

Once created, the CSR can be signed by Zebra or whichever PKI certificate authority is preferred.

Best Practices - MQTT Security

Certificates

By default, the printer comes supplied with a generic MQTT device certificate, but unlike Weblink does not contain a Zebra broker certificate authority.

A broker certificate authority must be provided to enable a successful connection via MQTT over TLS. Some brokers may also require username and password authentication, or specific device certificates that can be loaded on to the printer instead of using the generic default certificate.

Certificate Files

The following filenames shall be used to store the certificates:

```
MQTTX_CERT.NRD – device printer certificate
MQTTX_KEY.NRD – device printer private key (cannot be encrypted)
MQTTX_CA.NRD – broker certificate authority chain
```

Where "MQTTX" is either "MQTT1" or "MQTT2"

Retry Interval

To prevent flooding an MQTT broker with connections, it is recommended to configure a random retry interval. This allows for all the devices connecting to the MQTT broker to attempt reconnection at different times after a connection loss event. The SGD to configure this is:

```
mqtt.connX.retry_interval_random_max
```

Where `mqttX` is the connection 1 or 2 for MQTT

The printer will wait a random number of seconds between 1 and the value specified when attempting to reconnect.

Best Practices - Printer Time

Many certificates use time to ensure that the certificate is valid. The printer must also have the correct time set. If the printer is set to an earlier time than the certificate begins, or a later time than the certificate expires, the connection will be rejected. Additionally, having the correct time on the printer is useful for log event correlation.

Recommendation

The printer supports NTP configuration that will automatically set the printer time based on NTP server time using the following SGD commands:

```
ip.ntp.enable
ip.ntp.servers
```

If NTP is unavailable, manually set the printer time, using the following SGD commands:

```
rtc.time
rtc.date
```

Alternatively, you can also set the time using the standard Unix Epoch (number of seconds since January 1, 1970). Setting time in this manner is useful for devices that exist across multiple time zones. This can be configured using the following SGD command:

```
rtc.unix_timestamp
```

Best Practices - Printer Decommissioning

Starting in Link-OS 6, Zebra printers have a new capability - to delete all user data, reset all settings, and admin configurations. This feature also includes the option to wipe flash memory of any previous data with a maximum of 3 passes. The Decommissioning process provides the ability to know that sensitive data has been removed from the printer and it can be used for other purposes. It is also useful for restoring a printer back to configurability if a Protected Mode admin password is forgotten or lost.
To Decommission the printer, the user must specify the following ZPL command using the USB (client) connection:

```
~PM<printer serial number>,<number of flash wipe passes (default of 0)><CR>
```

For example:

```
~PM456c766973
```

Note: Decommissioning can only be performed using the USB (client) port. If the ~PM command is received on any other port, such as Bluetooth or Ethernet, it will be ignored.

This command would Decommission that printer only if the serial number matched what was specified in the command. The command will be ignored if the serial number does not match the printer, or if it was sent over any other port than USB.

Decommissioning a printer will remove the device from Protected Mode. If Protected Mode is being used, the printer will need to be placed back in that Mode after the Decommission is finished.

Note: There was an issue in Link-OS 6.0 that requires an additional sequence to be executed to finalize the Decommissioning Process. This issue was corrected in Link-OS 6.1, so the following only applies if using v6.0.

The required steps are:

1. Once the printer reboots, place the printer in Protected Mode, using this JSON command:

```
{}{
  "protect":{
      "authentication":{
      "username":"admin",
      "password":"",
      "type":"basic"
      },
      "operation":"setup",
      "setup":{
      "username":"admin",
      "password":"Ant1%oTdq$2P9f"
      }
  }
}
```

2. Then, exit Protected Mode, using the Password you previously used to enter Protected Mode. For example, if your password was “Ant1%oTdq$2P9f” as shown above, you’d send this JSON command to the printer:

```
{}{
   "protect":{
        "authentication":{
        "username":"admin",
        "password":"Ant1%oTdq$2P9f",
        "type":"basic"
        },
        "operation":"setup",
        "setup":{
        "username":"admin",
        "password":""
        }
   }
}
```

Recommendation: Only issue a flash wipe if the printer will be resold, recycled, or reused by another group that should not have access to the printer data. This may include proprietary fonts, formats, files, or network configuration. A flash wipe does take considerable time, which will vary in length, based on printer model.

Protected SGD Commands

The following Set/Get/Do (SGD) commands are affected by Protected Mode being introduced in Link-OS 6. Some SGD commands can affect other settings; these are called "Linked Commands".

For more information on the syntax and use of each command, please see the Programmers Guide.

Protected Command
apl.enable
bluetooth.allow_non_display_numeric_comparison bluetooth.bluetooth_pin bluetooth.bonding bluetooth.clear_bonding_cache bluetooth.discoverable bluetooth.enable bluetooth.enable_reconnect bluetooth.friendly_name bluetooth.json_config_channel_enable bluetooth.le.contoller_mode bluetooth.le.power_class bluetooth.minimum_security_mode bluetooth.power_class
capture.channel1.port
card.enable
device.allow_firmware_downloads device.fips.enabled device.friendly_name device.prompted_network_reset device.reset device.syslog.configuration device.syslog.enable device.xml.enable
display.password.current
input.capture
internal_wired.8021x.password internal_wired.8021x.peap.anonymous_identity internal_wired.8021x.peap.validate_server_certificate internal_wired.8021x.privkey_password internal_wired.8021x.security internal_wired.8021x.ttls_tunnel internal_wired.8021x.username internal_wired.enable internal_wired.ip.addr internal_wired.ip.arp_interval internal_wired.ip.default_addr_enable internal_wired.ip.dhcp.arp_verify internal_wired.ip.dhcp.cache_ip internal_wired.ip.dhcp.cid_all internal_wired.ip.dhcp.cid_enable internal_wired.ip.dhcp.cid_prefix internal_wired.ip.dhcp.cid_suffix

Protected Command
internal_wired.ip.dhcp.cid_type internal_wired.ip.dhcp.option12 internal_wired.ip.dhcp.option12_format internal_wired.ip.dhcp.option12_value internal_wired.ip.dns.domain internal_wired.ip.dns.servers internal_wired.ip.gateway internal_wired.ip.netmask internal_wired.ip.port (see ip.port linked command) internal_wired.ip.port_alternate (see ip.port.alternate linked command) internal_wired.ip.port_json_config (see ip.port.json.config linked command) internal_wired.ip.protocol internal_wired.ip.timeout.enable internal_wired.ip.timeout.value internal_wired.ip.wins.addr internal_wired.ip.wins.permanent_source
ip.bootp.enable (see wlan.ip.protocol linked command) ip.dhcp.auto_provision_enable ip.dhcp.enable (see wlan.ip.protocol linked command) ip.dhcp.ntp.enable ip.firewall.authentication.add ip.firewall.authentication.remove ip.firewall.proxy ip.firewall.whitelist_in ip.ftp.enable ip.ftp.execute_file ip.http.custom_link_name ip.http.custom_link_url ip.http.enable ip.http.faq_url ip.https.enable ip.lpd.enable ip.mirror.auto ip.mirror.password ip.mirror.username ip.ntp.enable ip.ntp.servers ip.pop3.enable ip.port (linked to internal.wired.ip.port) ip.port_alternate (linked to internal.wired.ip.port.alternate) ip.port_json_config (linked to internal.wired.ip.port.json.config) ip.port_single_conn ip.smtp.enable ip.snmp.enable ip.snmp.get_community_name ip.snmp.set_community_name ip.snmp.trap_community_name ip.tcp.enable ip.tls.enable ip.tls.port ip.tls.port_json_config ip.udp.enable

Protected Command
mqtt.conn1.clean_session_flag mqtt.conn1.mqtt_version mqtt.conn1.password mqtt.conn1.ping_interval mqtt.conn1.qos mqtt.conn1.retry_interval_random_max mqtt.conn1.server_address mqtt.conn1.tenant_id mqtt.conn1.username mqtt.conn2.clean_session_flag mqtt.conn2.mqtt_version mqtt.conn2.password mqtt.conn2.ping_interval mqtt.conn2.qos mqtt.conn2.retry_interval_random_max mqtt.conn2.server_address mqtt.conn2.tenant_id mqtt.conn2.username mqtt.enable mqtt.restore_defaults
rtc.date rtc.time rtc.timezone rtc.unix_timestamp
usb.host.lock_out usb.mirror.enable
weblink.ip.conn1.location weblink.ip.conn2.location weblink.zebra_connector.enable
wlan.8021x.authentication (see wlan.security linked command) wlan.8021x.eap.password (see wlan password linked command) wlan.8021x.eap.privkey_password (see wlan.private.key.password linked command) wlan.8021x.eap.username (see wlan.username linked command) wlan.8021x.enable (wlan.security) (see wlan.password linked command) wlan.8021x.peap.anonymous_identity (see wlan.username linked command) wlan.8021x.peap.peap_password (see wlan.private_key.password linked command) wlan.8021x.peap.peap_username wlan.8021x.peap.privkey_password wlan.8021x.peap.validate_server_certificate wlan.8021x.ttls_tunnel wlan.allowed_band wlan.channel_mask wlan.enable wlan.essid wlan.ip.addr wlan.ip.arp_interval wlan.ip.default_addr_enable wlan.ip.dhcp.arp_verify wlan.ip.dhcp.cache_ip wlan.ip.dhcp.cid_all wlan.ip.dhcp.cid_enable wlan.ip.dhcp.cid_prefix

Protected Command
wlan.ip.dhcp.cid_suffix wlan.ip.dhcp.cid_type wlan.ip.dhcp.option12 wlan.ip.dhcp.option12_format wlan.ip.dhcp.option12_value wlan.ip.dhcp.required wlan.ip.dns.domain wlan.ip.dns.servers wlan.ip.gateway wlan.ip.netmask wlan.ip.protocol (Linked to ip.bootp.enable and ip.dhcp.enable) wlan.ip.timeout.enable wlan.ip.timeout.value wlan.ip.wins.addr wlan.ip.wins.permanent_source wlan.leap_mode (see wlan.security linked command) wlan.leap_password (see wlan.password linked command) wlan.leap_username (see wlan.username linked command) wlan.operating_mode wlan.password (Linked to wlan.8021x.eap.password, wlan.8021x.enable, wlan.leap_password) wlan.private_key_password (Linked to wlan.8021x.eap.privkey_password) wlan.rts_cts_enabled wlan.secure_ssid wlan.security (Linked to wlan.8021x.authentication, wlan.leap_mode, wlan.wpa.authentication, wlan.wpa.enable) wlan.user_channel_list wlan.username (Linked to wlan.8021x.eap.username, wlan.8021x.peap.anonymous_identity, wlan.leap_username) wlan.wpa.authentication (see wlan.security linked command) wlan.wpa.enable (see wlan.security linked command) wlan.wpa.groupkey_ciphersuite wlan.wpa.pairwise_ciphersuite wlan.wpa.psk wlan.wpa.timecheck wlan.wpa.wpa_version
zbi.enable
zpl.label_length_always

JSON Commands Response Codes

Code	Meaning
0	Command completed successfully
100	Invalid or missing user name or password
101	Invalid user name or password
102	Command is protected, requested operation will not be taken
200	Unsupported operation
205	Requested operation is missing or not expressed as a string
300	Invalid setup section (missing user name or password)
301	Invalid user name
302	Password used is too short
303	Password used is too long
304	Password used invalid characters