

# **Programming Guide**

---

## **Addendum for Link-OS™**



## 2 | Copyright and Liability Disclaimer

© 2013 ZIH Corp. The copyrights in this manual and the software and/or firmware in the printer described therein are owned by ZIH Corp. and Zebra's licensors. Unauthorized reproduction of this manual or the software and/or firmware in the printer may result in imprisonment of up to one year and fines of up to \$10,000 (17 U.S.C.506). Copyright violators may be subject to civil liability.

This product may contain ZPL<sup>®</sup>, ZPL II<sup>®</sup>, and Link-OS<sup>™</sup> programs; Element Energy Equalizer<sup>™</sup> Circuit; E<sup>3™</sup>; and Monotype Imaging fonts. Software © ZIH Corp. All rights reserved worldwide.

Link-OS<sup>™</sup>, and all product names and numbers are trademarks, and Zebra, the Zebra head graphic, ZPL and ZPL II are registered trademarks of ZIH Corp. All rights reserved worldwide.

All other brand names, product names, or trademarks belong to their respective holders. For additional trademark information, please see "Trademarks" on the product CD.

**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.

### **rapidJson Library** Copyright © 2011 Milo Yip

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **libCurl Library** COPYRIGHT AND PERMISSION NOTICE Copyright © 1996 - 2012, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

---

# Contents

|   |           |
|---|-----------|
| Contacts .....  | 5         |
| <b>Using Weblink .....</b>  | <b>7</b>  |
| What is Weblink and when should it be used? .....                 | 7         |
| Configuring Weblink .....   | 8         |
| Basic Configuration .....   | 8         |
| When a Proxy Server is Part of the Network Configuration .....    | 8         |
| When HTTP Authentication is Necessary .....                       | 9         |
| Additional Firewall Configuration .....                           | 10        |
| Difference Between Conn1 and Conn2 .....                          | 10        |
| Enable Logging .....  | 10        |
| Navigating the Log Output .....                                   | 11        |
| SSL/TLS Certificate Errors .....                                  | 12        |
| Other Typical Errors .....  | 14        |
| Troubleshooting .....   | 15        |
| HTTP Messages .....   | 16        |
| <b>HTTP POST Alerts .....</b>                                     | <b>17</b> |
| Configuring Alerts Where the Alert Destination is HTTP POST ..... | 17        |
| How to Parse via PHP .....  | 19        |
| Basic Configuration .....   | 19        |
| When a Proxy Server is Part of the Network Configuration .....    | 20        |
| When HTTP Authentication is Necessary .....                       | 21        |
| Enabling Logging .....  | 21        |
| Navigating the Log Output .....                                   | 22        |
| Understanding Errors in the Alerts HTTP Log .....                 | 22        |
| Troubleshooting .....   | 23        |
| HTTP Messages .....   | 24        |

|   |           |
|---|-----------|
| <b>Set / Get / Do Commands</b>                        | <b>25</b> |
| Configuring JSON Usage for Communications             | 25        |
| alerts.add  | 27        |
| alerts.conditions                                     | 29        |
| alerts.configured                                     | 30        |
| alerts.destinations                                   | 31        |
| alerts.http.authentication.add                        | 32        |
| alerts.http.authentication.entries                    | 34        |
| alerts.http.authentication.remove                     | 35        |
| alerts.http.logging.clear                             | 36        |
| alerts.http.logging.entries                           | 37        |
| alerts.http.logging.max_entries                       | 38        |
| alerts.http.proxy                                     | 39        |
| alerts.tracked_settings.log_tracked                   | 41        |
| alerts.tracked_settings.clear_log                     | 42        |
| alerts.tracked_sgds.log                               | 43        |
| alerts.tracked_sgds.max_log_entries                   | 44        |
| alerts.tracked_sgds.zbi_notified                      | 45        |
| appl.link_os_version                                  | 46        |
| capture.channel1.count                                | 47        |
| capture.channel1.data.mime                            | 48        |
| capture.channel1.data.raw                             | 49        |
| capture.channel1.delimiter                            | 50        |
| capture.channel1.max_length                           | 51        |
| capture.channel1.port                                 | 52        |
| device.user_vars.create                               | 54        |
| device.user_vars.set_range                            | 56        |
| internal_wired.ip.port_json_config                    | 57        |
| ip.port_json_config                                   | 58        |
| weblink.enable  | 59        |
| weblink.ip.conn[1 2].authentication.add               | 60        |
| weblink.ip.conn[1 2].authentication.entries           | 62        |
| weblink.ip.conn[1 2].authentication.remove            | 63        |
| weblink.ip.conn[1 2].location                         | 64        |
| weblink.ip.conn[1 2].maximum_simultaneous_connections | 65        |
| weblink.ip.conn[1 2].proxy                            | 66        |
| weblink.ip.conn[1 2].retry_interval                   | 68        |
| weblink.ip.conn[1 2].test.location                    | 69        |
| weblink.ip.conn[1 2].test.retry_interval              | 71        |
| weblink.ip.conn[1 2].test.test_on                     | 72        |
| weblink.logging.clear                                 | 74        |
| weblink.logging.entries                               | 75        |
| weblink.logging.max_entries                           | 76        |
| weblink.printer_reset_required                        | 78        |
| weblink.restore_defaults                              | 79        |
| wlan.ip.port_json_config                              | 80        |

## Contacts

Technical Support via the Internet is available 24 hours per day, 365 days per year.

**Web Site:** [www.zebra.com](http://www.zebra.com)

**E-mail Back Technical Library:**

E-mail address: [emb@zebra.com](mailto:emb@zebra.com)

Subject line: Emaillist

**Self Service Knowledge Base:** [www.zebra.com/knowledgebase](http://www.zebra.com/knowledgebase)

**Online Case Registration:** [www.zebra.com/techrequest](http://www.zebra.com/techrequest)

| Which Department Do You Need?   | The Americas  | Europe, Middle East, and Africa   | Asia Pacific and India   |
|---|---|---|--|
| <b>Regional Headquarters</b>  | Zebra Technologies Corporation<br>475 Half Day Road, Suite 500<br>Lincolnshire, IL 60069 USA<br>T: +1 847 634 6700<br>Toll-free +1 866 230 9494<br>F: +1 847 913 8766   | Zebra Technologies Europe Limited<br>Dukes Meadow<br>Millboard Road<br>Bourne End<br>Buckinghamshire, SL8 5XF<br>United Kingdom<br>T: +44 (0) 1628 556000<br>F: +44 (0) 1628 556001                             | Zebra Technologies Asia Pacific Pte. Ltd.<br>120 Robinson Road<br>#06-01 Parakou Building<br>Singapore 068913<br>T: +65 6858 0722<br>F: +65 6885 0838  |
| <b>Technical Support</b><br>For questions on the operation of Zebra equipment and software, please call your distributor. For additional assistance, contact us. Please have your model and serial numbers available. | T: +1 877 ASK ZEBRA (275 9327)<br>F: +1 847 913 2578<br>Hardware: <a href="mailto:ts1@zebra.com">ts1@zebra.com</a><br>Software: <a href="mailto:ts3@zebra.com">ts3@zebra.com</a><br><i>Kiosk printers:</i><br>T: +1 866 322 5202<br>E: <a href="mailto:kiosksupport@zebra.com">kiosksupport@zebra.com</a> | T: +44 (0) 1628 556039<br>F: +44 (0) 1628 556003<br>E: <a href="mailto:Tseurope@zebra.com">Tseurope@zebra.com</a>   | T: +65 6858 0722<br>F: +65 6885 0838<br>E: China: <a href="mailto:tschina@zebra.com">tschina@zebra.com</a><br>All other areas:<br><a href="mailto:tsasiapacific@zebra.com">tsasiapacific@zebra.com</a>     |
| <b>Repair Service Department</b><br>For back-to-base service and repair.  | T: +1 877 ASK ZEBRA (275 9327)<br>F: +1 847 821 1797<br>E: <a href="mailto:repair@zebra.com">repair@zebra.com</a><br>To request a repair in the U.S., go to <a href="http://www.zebra.com/repair">www.zebra.com/repair</a> .  | T: +44 (0) 1772 693069<br>F: +44 (0) 1772 693046<br>New requests: <a href="mailto:ukrma@zebra.com">ukrma@zebra.com</a><br>Status updates:<br><a href="mailto:repairupdate@zebra.com">repairupdate@zebra.com</a> | T: +65 6858 0722<br>F: +65 6885 0838<br>E: China: <a href="mailto:tschina@zebra.com">tschina@zebra.com</a><br>All other areas:<br><a href="mailto:tsasiapacific@zebra.com">tsasiapacific@zebra.com</a>     |
| <b>Technical Training Department</b><br>For Zebra product training courses.   | T: +1 847 793 6868<br>T: +1 847 793 6864<br>F: +1 847 913 2578<br>E: <a href="mailto:tamerica@zebra.com">tamerica@zebra.com</a>   | T: +44 (0) 1628 556000<br>F: +44 (0) 1628 556001<br>E: <a href="mailto:Furtraining@zebra.com">Furtraining@zebra.com</a>   | T: +65 6858 0722<br>F: +65 6885 0838<br>E: China: <a href="mailto:tschina@zebra.com">tschina@zebra.com</a><br>All other areas:<br><a href="mailto:tsasiapacific@zebra.com">tsasiapacific@zebra.com</a>     |
| <b>Inquiry Department</b><br>For product literature and distributor and dealer information.   | T: +1 877 ASK ZEBRA (275 9327)<br>E: <a href="mailto:inquiry4@zebra.com">inquiry4@zebra.com</a>   | T: +44 (0) 1628 556037<br>F: +44 (0) 1628 556005<br>E: <a href="mailto:mseurope@zebra.com">mseurope@zebra.com</a>   | E: China: <a href="mailto:GCmarketing@zebra.com">GCmarketing@zebra.com</a><br>All other areas:<br><a href="mailto:APACChannelmarketing@zebra.com">APACChannelmarketing@zebra.com</a>                       |
| <b>Customer Service Department (US)</b><br><b>Internal Sales Department (UK)</b><br>For printers, parts, media, and ribbon, please call your distributor or contact us.   | T: +1 877 ASK ZEBRA (275 9327)<br>E: <a href="mailto:clientcare@zebra.com">clientcare@zebra.com</a>   | T: +44 (0) 1628 556032<br>F: +44 (0) 1628 556001<br>E: <a href="mailto:cseurope@zebra.com">cseurope@zebra.com</a>   | T: +65 6858 0722<br>F: +65 6885 0836<br>E: China: <a href="mailto:order-csr@zebra.com">order-csr@zebra.com</a><br>All other areas:<br><a href="mailto:csasiapacific@zebra.com">csasiapacific@zebra.com</a> |

**Key:** T: Telephone  
F: Facsimile  
E: E-mail



**Notes •** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

---

# Using Weblink

## What is Weblink and when should it be used?

Weblink is a feature of Zebra Link-OS™ printers. Using a secure connection, the Weblink feature allows the printer to directly connect to an internet based server, for the purpose of either sending information to the server or receiving from the server. Weblink can transport data securely through a firewall.

Weblink can transport any information related to device management, transactional data and information to be processed at a later time. It can be used as part of an overall cost reduction solution that leverages web technologies.

Typically, an application called a ‘servlet’ is created and run on the internet based server, waiting for printers to connect and interact with the servlet. These servlet applications can provide a variety of functions – from sending operating system updates to the printer, to receiving data from the printer and, in turn, using that data to trigger events in other systems.

For example, a solution could be created that would feature the printer consuming data from a Bluetooth® scanner connected to the printer – with that scanned data then being sent from the printer to the internet-based servlet. The servlet would then seek out additional details related to the scanned data, format a document, and then send it to the printer for printing.

## Configuring Weblink

When any Weblink setting (with the exception of the logging settings) is adjusted either via SNMP, SGD, or JSON it is required that the printer be reset before the new value takes effect. The `weblink.printer_reset_required` setting will be set to "yes" if there are any settings that have been modified that require a printer reset.

### Basic Configuration

To determine how much configuration is necessary the following questions should be considered:

1. Is the remote server the printer is attempting to connect to outside the corporate firewall?
2. Does the firewall require a username and password to access the remote server?
3. Does the printer require a proxy server to access the remote server?
4. Does the firewall permit HTTPS connections initially or does the printer need to connect via HTTP first?

If the answer to any of these questions is 'yes', then more than the basic configuration may be necessary. Depending upon the network environment that the printer is in, accessing the remote server may only require that a few settings be set.

The minimum requirement is that the URL for the remote server be set. For simplicity, assume that only conn1 is being used (this is the typical scenario). See also [Difference Between Conn1 and Conn2 on page 10](#).

To configure the printer to connect to the remote server:

- a. Set `weblink.ip.conn1.location` to the URL of the remote server.  
The URL must conform to the standards described in RFC3986 (<http://www.ietf.org/rfc/rfc3986.txt>). For example, if the remote servlet's full URL is `https://www.examplecorpinc.com/zebra/weblink/`  
Configure the location setting as follows:  

```
! U1 setvar "weblink.ip.conn1.location"  
"https://www.examplecorpinc.com/zebra/weblink/"
```
- b. Reset the printer.  
When the printer has an IP address, it will attempt to connect to the remote server. In the event that the remote server does not indicate that the printer has connected, logging may need to be enabled in order to determine the failure.

### When a Proxy Server is Part of the Network Configuration

If a proxy server must be used to access the remote server, the printer's proxy setting must be set to connect to the server. There are typically four properties associated with a proxy server:

- The proxy server scheme: `HTTP` or `HTTPS`
- The proxy server address
- The proxy server port (optional)
- The username and password for the proxy (optional)



To supply the address of the proxy server (assuming a default port and no username/password), configure the proxy setting as follows:

```
! U1 setvar "weblink.ip.conn1.proxy" "https://my.internal.proxy/"
```

In this scenario the proxy address is `my.internal.proxy` and the scheme is `HTTPS`. The default port (1080) will be used. No username or password will be used to authenticate with the proxy.

To specify an alternate port configure the proxy as follows:

```
! U1 setvar "weblink.ip.conn1.proxy" "https://my.internal.proxy:3128/"
```

To specify a username and password configure the proxy as follows:

```
! U1 setvar "weblink.ip.conn1.proxy" "https://user:pass@my.internal.proxy/"
```

The proxy username, password, and the rest of the URL must follow the rules specified in RFC3986 (<http://www.ietf.org/rfc/rfc3986.txt>).

## When HTTP Authentication is Necessary

Use this configuration when, for example, a firewall requires a username and/or password.

It may be necessary to specify a username and password to various routers and servers along the path to the remote server. Typically when using a browser to access the server the authentication request will be presented in the form of a dialog window that asks for the username and password.

Since the printer's connection to the remote server is headless and non-interactive, the Weblink configuration allows a user to enter in a server name/username/password triplet. The triplet will be used in the event that the printer is presented with an authentication request (for example, this typically is requested via the `HTTP/1.1 401 Unauthorized` request).

To specify authentication credentials, issue the following:

```
! U1 setvar "weblink.ip.conn1.authentication.add" "servername.com username  
password"
```

In this scenario the server requesting authentication is `servername.com`. The username and password to be supplied are 'username' and 'password'. The server name can be either a DNS name or an IP address. The username and password cannot be retrieved from SGD, SNMP, or JSON once added. Only the server name will be returned.

More than one set of authentication triplets can be added. The printer will only use the credentials as they are needed. In other words, the printer will only use the credentials for `servername.com` if it receives a `HTTP/1.1 401 Unauthorized` request from `servername.com`.

To see what authentication triplets are specified issue:

```
! U1 getvar "weblink.ip.conn1.authentication.entries"
```

To remove authentication credentials issue the following:

```
! U1 setvar "weblink.ip.conn1.authentication.remove" "servername.com"
```

## Additional Firewall Configuration

Some firewalls do not allow the first connection attempt for a device to be HTTPS or require new connections periodically to keep the initial connections intact. The weblink test branch was provided to address issues that typically arise because the printer is an unattended device.

To configure the printer to attempt an HTTP connection anytime the HTTPS connection drops :

```
! U1 setvar "weblink.ip.conn1.test.location"  
"http://www.zebra.com/apps/linktest"  
! U1 setvar "weblink.ip.conn1.test.test_on" "failure"
```

The `weblink.ip.conn1.test.location` can be any valid HTTP address. The default uses a link provided by Zebra that exists for no other purpose than to help developers test their connections to the internet. Setting `weblink.ip.conn1.test.test_on` to `interval` or `both` will force the printer to attempt a connection to the URL in location every `weblink.ip.conn1.test.retry_interval` seconds (default is 900 seconds/15 minutes).

To configure the printer to try an HTTP connection periodically, independent of the HTTPS success:

```
! U1 setvar "weblink.ip.conn1.test.location"  
"http://www.zebra.com/apps/linktest"  
! U1 setvar "weblink.ip.conn1.test.test_on" "interval"  
! U1 setvar "weblink.ip.conn1.test.retry_interval" "900"
```

## Difference Between Conn1 and Conn2

The printer has the ability to connect to two different servers. Connection 1 (`conn1`) and Connection 2 (`conn2`) are identical in every way in terms of their configuration. It is expected that `conn2` will typically be left unmodified unless a user has an alternate server that they wish to use to configure the printer.

A typical scenario in which both connections would be used is if a user wishes to have the printer connect to both a configuration server and a data source.

## Enable Logging

If your printer has trouble connecting, you may wish to enable logging. By default logging is not enabled in order to reduce the amount of memory consumed when the Weblink feature is enabled. It is recommended that once the Weblink feature is configured properly and is performing as expected that the logging be disabled or that a very small (less than 100) number of logging entries be permitted.

To enable logging, `weblink.logging.max_entries` needs to be modified. By default it is set to 0, which indicates that no messages are logged. When attempting to troubleshoot connection issues it is recommended that `max_entries` be set to at least 100 entries. Setting `max_entries` to 100 means that the 100 newest logging entries will be present in `weblink.logging.entries` as older entries are discarded once the maximum number of entries is reached.

```
! U1 setvar "weblink.logging.max_entries" "100"
```

The logging settings are atypical to the Weblink settings as they do not require the printer to be reset before taking effect. This does not mean that previous logging message that would have been logged will appear when the `max_entries` setting is changed from 0 to a greater value. It means that any new logging messages will be logged from that point forward.

Issue the following command to clear any log entries currently in the `weblink.logging.entries` buffer.

```
! U1 do "weblink.logging.clear" ""
```

## Navigating the Log Output

The log will contain useful information, even in the scenario where the printer successfully connects to the remote server. This section explains how to read the log and highlights some of the key entries that will help to determine if the connection was successful.

A typical log entry looks as follows:

```
[12-04-2012 14:57:10.625] [conn1.1] Attempting connection to  
https://www.examplecorpinc.com/zebra/weblink/
```

The first column is the date and time that the event occurred. The format of the date and time matches the format of `rtc.date` and `rtc.time`. The time, however, also includes the milliseconds to aid in troubleshooting network latency concerns.



**Note** • For printers that do not have a battery to store the Real Time Clock (RTC) value, the date will be restored to the default value upon a power cycle. The default value depends upon how the `rtc.date` SGD is managed. If it has never been set then it will default to the firmware build date (the value in `appl.date`). Otherwise, the value in `rtc.date` will default to the value that it was last set to. This does not mean the value of the `rtc.date` when it was power cycled. It means that when a user sets `rtc.date` that becomes the new default value.

If the printer has a battery then the `rtc.date` is never default and continues to track the date as expected.

The second column indicates the connection name and channel that the entries are associated with. The connection name will match the weblink branch that was configured with the respective URL (for example, `conn1` or `conn2`). The channel number indicates which channel on the respective connection the entries corresponds to.




**Note** • Channels are additional connections that are requested by the server when the server needs to perform a specific operation that cannot be done on the channel(s) currently open. Typically only the RAW channel is open which operates similar to the RAW TCP port. It is typical to see two channels opened, the main channel and the RAW channel.


The third column is the actual message that contains information about what occurred in the printer at the corresponding time in column one. In the above example the printer was initiating the connection to the URL specified in `weblink.ip.conn1.location`.

Review the section titled [Understanding Errors in the Weblink Log on page 11](#) to understand what it means when certain logging messages/errors appear in the log.

## SSL/TLS Certificate Errors


Secure connections to the remote server present the opportunity for several errors when attempting to connect. The errors typically involve the certificates used when connecting via SSL or TLS. This section highlights some of the most common issues involving the certificates.

| Error   | Cause / Solution   |
|---|--|
| "SSL certificate problem: self signed certificate in certificate chain" | One of the situations that prevent a successful connection is not having the correct Certificate Authority certificates installed on the remote server. Zebra requires that the Zebra Root Certificate Authority and the Zebra Subordinate Certificate Authority be installed on the remote server. This error typically indicates that only one of the Zebra Certificate Authority certificates is installed on the remote server.  |
| "SSL certificate problem: unable to get local issuer certificate"       | One of the situations that prevent a successful connection is not having the correct Certificate Authority certificates installed on the remote server. Zebra requires that the Zebra Root Certificate Authority and the Zebra Subordinate Certificate Authority be installed on the remote server. This error typically indicates that neither of the Zebra Certificate Authority certificates are installed on the remote server.  |
| "SSL certificate problem: certificate has expired"                      | <p>This error indicates that the remote server's certificate has expired. This is typically an indication that the printer's date and/or time are incorrect as the Zebra certificates are typically issued for long durations. Check that <code>rtc.date</code> and <code>rtc.time</code> are set correctly.</p> <p> <b>Note</b> • For printers that do not have a battery to store the Real Time Clock (RTC) value, the date will be restored to the default value upon a power cycle. The default value depends upon how the <code>rtc.date</code> SGD is managed. If it has never been set then it will default to the firmware build date (the value in <code>appl.date</code>). Otherwise, the value in <code>rtc.date</code> will default to the value that it was last set to. This does not mean the value of the <code>rtc.date</code> when it was power cycled. It means that when a user sets <code>rtc.date</code> that becomes the new default value.</p> <p>If the printer has a battery then the <code>rtc.date</code> is never default and continues to track the date as expected.</p> |

| Error   | Cause / Solution  |
|---|---|
| "SSL certificate problem:<br>certificate is not yet valid"  | <p>This error indicates that the remote server's certificate was incorrectly issued or that the printer's date and/or time are incorrect. Check that the printer's date and time (<code>rtc.date</code> and <code>rtc.time</code>) are set correctly and that the certificate's start and expiration date are valid.</p> <p> <b>Note</b> • For printers that do not have a battery to store the Real Time Clock (RTC) value, the date will be restored to the default value upon a power cycle. The default value depends upon how the <code>rtc.date</code> SGD is managed. If it has never been set then it will default to the firmware build date (the value in <code>appl.date</code>). Otherwise, the value in <code>rtc.date</code> will default to the value that it was last set to. This does not mean the value of the <code>rtc.date</code> when it was power cycled. It means that when a user sets <code>rtc.date</code> that becomes the new default value. If the printer has a battery then the <code>rtc.date</code> is never default and continues to track the date as expected.</p> |
| "subjectAltName does not match<br>1.2.3.4"  | <p>Part of the certificate validation process involves making sure that the remote server is who it claims to be. A certificate can be created to validate against several aliases/DNS names. Typically the certificate will not contain the IP address of the server as IP addresses are subject to change. When specifying the remote server's URL via <code>weblink.ip.conn1.location</code> be sure to specify one of the DNS aliases listed in the certificate. The valid names will be listed either under the Common Name (CN) field and/or the subjectAltName (SAN or Subject Alternate Name) field within the certificate. For example, the certificate may have the CN set to <code>'examplecorpinc'</code> and the SAN set to <code>'examplecorpinc.com'</code> or <code>'alias.for.examplecorpinc.com'</code>. Any of the CN or SAN names can be used, but, as the IP address is not listed in the CN or SAN it cannot. It is not recommended that the IP address be part of the SAN if a DNS name is available to avoid connection issues that may arise due to subnet change or DHCP lease expirations, etc.</p>  |
| "SSL certificate subject name<br>'examplecorpinc.com' does not match<br>target host name '1.2.3.4'" |   |
| "Unknown SSL protocol error in<br>connection to ..."  | <p>When this message is seen it means that the remote server's SSL/TLS configuration is incorrect. Refer to <a href="#">Troubleshooting on page 15</a> to ensure the server and printer are both configured correctly.</p>  |
| I do not see any of these errors, but the<br>printer still does not connect.                        | <p>Refer to <a href="#">Troubleshooting on page 15</a> to ensure the server and printer are both configured correctly.</p>  |

## Other Typical Errors

While SSL/TLS connection errors are the most common, there are issues that can arise that prevent a successful connection. This section highlights the most common issues.

| Error   | Cause / Solution   |
|---|--|
| <p>"Read failed with an unexpected error"</p>                       | <p>This message typically indicates that connection to the remote server was lost. The connection can either be lost due to the server powering off or resetting, the firewall or proxy server shutting down the connection, or because the remote server gracefully requests that the connection be discontinued.</p> <p> <b>Note</b> • After 60 seconds of inactivity on the connection the printer will attempt to contact the server via a TCP Keepalive. If the connection is still present the server will respond and the connection will remain open. After 10 successive failed attempts to contact the remote the printer will assume the connection is severed and close the connection. The printer will resume it's attempt to connect to the remote server so that when the server comes back online the printer will re-establish communication.</p> |
| <p>"Failed to connect (SP = #, CI = #, UW = #, AC = #, PC = #)"</p> | <p>If this error is seen one or more of the '#' values will be set to 0. This is an indication of an incorrect configuration of the remote server. Ensure that the remote server is setup according to the Servlet configuration in the Zebra Link-OS SDK documentation.</p> <p>This typically indicates an incorrect version of the remote Application Server (for example, Apache/Tomcat version may be incorrect). If this issue persists contact Zebra Technical Support.. See <i>Contacts on page xxxii</i> and provide the output of the following command (ensure that logging is enabled and that this error appears within the entries).</p> <pre>! U1 getvar "weblink"</pre>   |

## Troubleshooting

Whenever troubleshooting a connection issue, the following questions should be answered to ensure the configuration is correct.

1. Is the printer connected correctly via Wireless or Ethernet?
2. Does the printer have a valid IP address?
3. Can I ping the printer's IP address from a device on the same network as the printer?
4. Is the remote server URL in `weblink.ip.conn1.location` correct and does it point to the remote server that is configured for weblink functionality?
5. Can you connect to the location defined in the `weblink.ip.conn1.location` setting via a browser?
6. Is the remote server I am attempting to connect to outside the corporate firewall?
7. Can the URL specified in `weblink.ip.conn1.test.location` be accessed?  
If this is the case, talk with your administrator about altering restrictions for accessing HTTPS connections.
8. Does the firewall require a username and password to access the remote server?
9. Do I require a proxy server to access the remote server?
10. Is the proxy server port the default (1080) or another port (for example, 3128 for the Linux Squid proxy)?



**Note** • If using the Linux Proxy Server Squid, and you are having trouble connecting, note that it may be configured to:

- a. disallow POST messages
- b. only operate in HTTP/1.0 mode 3
- c. disallow SSL connections.

Refer to your Linux Squid documentation for complete details.

11. Does the firewall permit HTTPS connections initially or do I need to connect via HTTP first?
12. Is the remote server configured to use SSL v3.0 or TLS 1.0?  
TLS1.0 is recommended, but SSL v3.0 is also supported. Versions prior to SSL v3.0 and later than TLS v1.0 are not currently supported.
13. Are the Zebra Certificate Authority Certificates correctly installed on the remote server?
14. Was the server's certificate issued by Zebra and is it signed by the Zebra Certificate Authority?
15. Has the server's certificate expired?
16. Is the printer's date and time within the issue and expired period of the server's certificate?

17. Does the value in `weblink.ip.conn1.location` match either the Common Name or one of the names listed in the Subject Alternate Name of the remote server's certificate?
18. Is the proxy server configured correctly and does the respective proxy server allow HTTPS connections via the HTTP CONNECT method?
19. Are there any HTTP authentication attempts when trying to connect that fail?
20. Are there any HTTP/1.1 4xx messages in the log?

If your connection issues persist and the solutions in this document do not help, contact Zebra Tech Support and provide the output of the following command. Ensure that logging is enabled and that the error(s) appear within the entries)

```
! U1 getvar "weblink"
```

## HTTP Messages

| Message                          | Cause / Solution  |
|----------------------------------|---|
| HTTP/1.1 100 Continue            | This indicates that the server and printer have begun communicating and is often seen in place of HTTP/1.1 200 OK.  |
| HTTP/1.1 101 Switching Protocols | This indicates that the basic connection to the server worked and the protocol is being switched to a more efficient protocol for data transfer.  |
| HTTP/1.1 200 OK                  | This indicates that an HTTP GET or HTTP POST was successful.  |
| HTTP/1.1 30x Moved/Redirect/etc  | This indicates that the URL specified has moved or that the firewall redirected the printer to another location (typically this is done to authenticate a user in a transparent proxy configuration). |
| HTTP/1.1 401 Unauthorized        | This indicates that the printer either needs to authenticate with the server or failed to authenticate with the remote server (or server/router along the route to the server).                       |
| HTTP/1.1 403 Forbidden           | This typically means that the authentication was provided and valid, however, the user does not have access to the requested resource.  |
| HTTP/1.1 404 Not Found           | This indicates that the remote URL provided points to an invalid location on the server. This does indicate, however, that the server name is valid. Just the path after the domain name is invalid.  |



---

# HTTP POST Alerts

Link-OS printers can issue alerts to a web server that is listening for HTTP POST requests. The advantage of an HTTP POST alert over the other destinations available (for example, TCP, UDP, SNMP) is that HTTP is firewall friendly.

## Configuring Alerts Where the Alert Destination is HTTP POST

Any setting in the `alerts.http` branch that is set will take effect for any HTTP POST alerts that occur from that point forward. A printer reset is not required for the settings to take effect.

If the server is configured to accept and process HTTP POST messages either via a CGI script or a server-side script such as PHP or ASP then the alert can be forwarded to that server from the printer. The printer will send the alert using the multipart/form-data Content-Type. This allows any type of data, including binary data, to be sent via the POST.

The POST will support two variables within the body of the POST:

- `alertMsg` – This is the alert details and the content follows the format of a standard alert when it is issued over one of the other alert destinations (for example, serial, USB, TCP, etc.).

- `uniqueId` – The unique id of the printer. This matches the value in `device.unique_id`

The HTTP POST request will look as follows (sent when the printer was paused)

```
POST /http_post/alert.php HTTP/1.1
Host: 10.3.4.58
Accept: */*
Connection: close
Content-Length: 281
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----350c75835f46

-----350c75835f46
Content-Disposition: form-data; name="alertMsg"

ALERT%3A%20PRINTER%20PAUSED
-----350c75835f46
Content-Disposition: form-data; name="uniqueId"

XXQLJ120900310
-----350c75835f46--
```



**Important** • The message is using HTTP/1.1 and therefore HTTP/1.1 header fields. This is important because some older proxy servers do not handle these fields gracefully and may block the `POST` message.

It is important to note that the message is using HTTP/1.1 and therefore HTTP/1.1 header fields. This is important because some older proxy servers do not handle these fields gracefully and may block the `POST` message.

## How to Parse via PHP

The following example shows how to parse the POST message. It does not, however, show how to use this information on other pages, store the results in a database, report this to another device on the domain, etc. The response in this example will be sent back to the printer, but it will be ignored by the printer. If you wish to see the response you can use a packet sniffing tool such as Wireshark.

```
<?php

    $alertMsg = urldecode($_POST["alertMsg"]);
    if (preg_match("/(\w+(\s+\w+)?):\s+(((SGD SET)\s+([\w\d\.\_]+)\s+>\s+(.+))|([\w|\s]+))/", $alertMsg, $matches)== 1) {

        $alertType = $matches[1];

        if ($matches[5] === "SGD SET") {
            $alertCondition = $matches[5];
            $alertSgdName = htmlspecialchars($matches[6]);
            $alertSgdData = htmlspecialchars($matches[7]);
        } else {
            $alertCondition = $matches[3];
        }
    }

    echo "<H1>Alert Received</H1><br/>\r\n";
    echo "<h2>Original Message = $alertMsg</h2><br/>\r\n";
    echo "<h2>Alert Type = $alertType</h2><br/>\r\n";
    echo "<h2>Alert Condition = $alertCondition</h2><br/>\r\n";
    echo "<h2>SGD Name = $alertSgdName</h2><br/>\r\n";
    echo"<h2>SGD Value = $alertSgdData</h2><br/>\r\n";

?>
```

## Basic Configuration

To determine how much configuration is necessary the following questions should be considered:

1. Is the remote server the printer is attempting to connect to outside the corporate firewall?
2. Does the firewall require a username and password to access the remote server?
3. Does the printer require a proxy server to access the remote server?

If the answer to any of these questions is ‘yes’, then more than the basic configuration may be necessary. Depending upon the network environment that the printer is in access the remote server may only require that a few settings be set.

To configure an alert to be sent via HTTP POST to a remote server the following setting should be set as follows:

```
! U1 setvar "alerts.add" "PRINTER PAUSED,HTTP-
POST,Y,Y,http://www.examplecorpinc.com/alerts.php,0,N,"
```

The above will issue an HTTP POST alert to the remote server (<http://www.examplecorpinc.com/alerts.php>) when the printer is paused or un-paused.

The first parameter indicates the condition to monitor. A list of available alert conditions can be viewed by issuing:

```
! U1 getvar "alerts.conditions"
```

The second parameter indicates the alert destination. For the purposes of this section HTTP-POST is the preferred destination. A list of available alert destinations can be viewed by issuing:

```
! U1 getvar "alerts.destinations"
```

The third and fourth parameters are ‘Send on Set’ and ‘Send on Clear’, respectively. They can be either "Y" for monitor the alert or "N" for don't monitor the alert. If both are set to "N" then the alert will not be added or it will be deleted if the alert already existed. To view which alerts already exist issue:

```
! U1 getvar "alerts.configured"
```

The fifth parameter holds the URL for the server that will be sent the HTTP POST. It holds a maximum of 255 characters for the URL and it must conform to the URI standards described in RFC3986 (<http://www.ietf.org/rfc/rfc3986.txt>).

The sixth parameter should be set to 0 for HTTP POST alerts.

The seventh parameter and eighth parameter will not be covered in this section and should be set as indicated in the description above. See the SGD documentation for details on these two parameters.

## When a Proxy Server is Part of the Network Configuration

If a proxy server must be used to access the remote server the printer's proxy setting must be set to point to the server. There are typically four properties associated with a proxy server.

- The proxy server scheme: HTTP is the only supported scheme
- The proxy server address
- The proxy server port (optional)
- The username and password for the proxy (optional)

To supply the address of the proxy server, assuming a default port and no username/password, configure the proxy setting as follows:

```
! U1 setvar "alerts.http.proxy" "http://my.internal.proxy/"
```

In this scenario the proxy address is `my.internal.proxy` and the scheme is `HTTP`. The default port (1080) will be used. No username or password will be used to authenticate with the proxy.

To specify an alternate port configure the proxy as follows

```
! U1 setvar "alerts.http.proxy" "http://my.internal.proxy:3128/"
```

To specify a username and password configure the proxy as follows:

```
! U1 setvar "alerts.http.proxy" "http://user:pass@my.internal.proxy/"
```

The proxy username, password, and the rest of the URL must follow the rules specified in RFC3986 (<http://www.ietf.org/rfc/rfc3986.txt>).

## When HTTP Authentication is Necessary

Use this method when, for example, a firewall requires a username and password.

It may be necessary to specify a username and password to various routers and servers along the path to the remote server. Typically when using a browser to access the server the authentication request will be presented in the form of a dialog window that asks for the username and password. As the printer's connection to the remote server is headless and non-interactive the alert http configuration allows a user to enter in a server name, username, password triplet. The triplet will be used in the event that the printer is presented with an authentication request (for example, this typically is requested via the HTTP/1.1 401 Unauthorized request).

To specify authentication credentials issue the following:

```
! U1 setvar "alerts.http.authentication.add" "servername.com username password"
```

In this scenario the server requesting authentication is `servername.com`. The username and password to be supplied are 'username' and 'password'. The server name can be either a DNS name or an IP address. The username and password cannot be retrieved from SGD, SNMP, or JSON once added. Only the server name will be returned.

More than one set of authentication triplets can be added. The printer will only use the credentials as they are needed. In other words, the printer will only use the credentials for `servername.com` if it receives a HTTP/1.1 401 Unauthorized request from `servername.com`.

To see what authentication triplets are specified issue:

```
! U1 getvar "alerts.http.authentication.entries"
```

To remove authentication credentials issue the following:

```
! U1 setvar "alerts.http.authentication.remove" "servername.com"
```

## Enabling Logging

If your printer has trouble connecting, you may wish to enable logging. By default logging is not enabled in order to reduce the amount of memory consumed when the HTTP alert feature is enabled. It is recommended that once the alert HTTP feature is configured properly and is performing as expected that the logging be disabled or that a very small (less than 100) number of logging entries be permitted.

To enable logging, `alerts.http.logging.max_entries` needs to be modified. By default it is set to 0, which indicates that no messages are logged. When attempting to troubleshoot connection issues it is recommended that `max_entries` be set to at least 100 entries. Setting `max_entries` to 100 means that the 100 newest logging entries will be present in `alerts.http.logging.entries` as older entries are discarded once the maximum number of entries is reached.

```
! U1 setvar "alerts.http.logging.max_entries" "100"
```

The logging settings do not require the printer to be reset before taking effect. This does not mean that previous logging message that would have been logged will appear when the `max_entries` setting is changed from 0 to a greater value. It means that any new logging messages will be logged from that point forward.

Issue the following command to clear any log entries currently in the `alerts.http.logging.entries` buffer.

```
! U1 do "alerts.http.logging.clear" ""
```

## Navigating the Log Output

The log can contain much information, even in the scenario where the printer successfully connects to the remote server. This section explains how to read the log and highlights some of the key entries that will help to determine if the connection was successful.

A typical log entry looks as follows:

```
[12-04-2012 14:57:10.625] [http] Attempting connection to  
http://www.examplecorpinc.com/alerts.php
```

The first column is the date and time that the event occurred. The format of the date and time matches the format of `rtc.date` and `rtc.time`. The time, however, also includes the milliseconds to aid in troubleshooting network latency concerns.

The second column indicates the connection name, which is currently always set to 'http'.

The third column is the actual message that contains information about what occurred in the printer at the corresponding time in column one. In the above example the printer was attempting to POST the alert to the connection to the URL specified in the configured alert.

Review [Understanding Errors in the Alerts HTTP Log on page 22](#) to understand what it means when certain logging messages/errors appear in the log.

## Understanding Errors in the Alerts HTTP Log

| Error                    | Cause / Solution   |
|--------------------------|--|
| Couldn't connect to host | <p>This could mean any number of things occurred that prevented the printer from connecting. This message is always present when the connection to the remote failed and is typically accompanied by an HTTP Response Code. See <a href="#">HTTP Messages on page 24</a> for the possible HTTP Response Codes and their meaning.</p> <p>If this issue persists contact Zebra Technical Support.. See <a href="#">Contacts on page xxxii</a> and provide the output of the following command (ensure that logging is enabled and that this error appears within the entries).</p> <pre>! U1 getvar "alerts"</pre> |

## Troubleshooting

Whenever troubleshooting a connection issue, the following questions should be answered to ensure the configuration is correct.

1. Is the printer connected correctly via Wireless or Ethernet?
2. Does the printer have a valid IP address?
3. Can I ping the printer's IP address from a device on the same network as the printer?
4. Is the remote server URL listed in the output of the `alerts.configured` SGD or `~HU` command output correct, and is the URL of a server iserver that is configured to accept HTTP POST requests?
5. Can you connect to the URL via a browser?
6. Is the remote server I am attempting to connect to outside the corporate firewall?
7. Does the firewall require a username and password to access the remote server?
8. Does the printer require a proxy server to access the remote server?
9. Is the proxy server port the default (1080) or another port?
10. Is the proxy server configured to allow HTTP POST messages?
11. Is the proxy server HTTP 1.1 compliant and does it allow HTTP 100 Continue messages?  
For example, the Squid Proxy Server versions before v3.2 do not fully support HTTP 1.1 requests and may block any HTTP POST attempts from the printer.



**Note** • If using the Linux Proxy Server Squid, and you are having trouble connecting, note that it may be configured to:

- a. disallow POST messages
- b. only operate in HTTP/1.0 mode 3
- c. disallow SSL connections.

Refer to your Linux Squid documentation for complete details.

12. Are there any HTTP authentication attempts when trying to connect that fail?

Are there any HTTP/1.1 4xx messages in the log?

If your connection issues persist and the solutions in this document do not help, contact Zebra Technical Support.. See *Contacts on page xxxii* and provide the output of the following command (ensure that logging is enabled and that this error appears within the entries).

```
! U1 getvar "alerts"
```

## HTTP Messages

| Message                         | Cause / Solution  |
|---------------------------------|---|
| HTTP/1.1 100 Continue           | This indicates that the server and printer have begun communicating and is often seen in place of HTTP/1.1 200 OK.  |
| HTTP/1.1 200 OK                 | This indicates that the HTTP POST was successful.   |
| HTTP/1.1 30x Moved/Redirect/etc | This indicates that the URL specified has moved or that the firewall redirected the printer to another location (typically this is done to authenticate a user in a transparent proxy configuration). |
| HTTP/1.1 401 Unauthorized       | This indicates that the printer either needs to authenticate with the server or failed to authenticate with the remote server (or server/router along the route to the server).                       |
| HTTP/1.1 403 Forbidden          | This typically means that the authentication was provided and valid, however, the user does not have access to the requested resource.  |
| HTTP/1.1 404 Not Found          | This indicates that the remote URL provided points to an invalid location on the server. This does indicate, however, that the server name is valid. Just the path after the domain name is invalid.  |



---

# Set / Get / Do Commands

## Configuring JSON Usage for Communications

To configure JSON usage for communication, refer to the following examples using `device.location` as an example.

➔ **Example 1** • To change `device.location` using SGD you do:

```
! U1 setvar "device.location" "my desk"
```

To do the same thing in JSON you do:

```
{ }{"device.location":"my desk"}
```

The response is:

```
{"device.location":"my desk"}
```

➔ **Example 2** • To do a `getvar` in SGD you do:

```
! U1 getvar "device.location"
```

In JSON:

```
{ }{"device.location":null}
```

The response is:

```
{"device.location":"my desk"}
```

➔ **Example 3** • You can get several values as follows:

```
{ }{"device.friendly_name":null, "device.company_name":null,  
"device.company_contact":null, "device.location":null}
```

The response is:

```
{"device.friendly_name":"XXQLJ120900310",  
"device.company_name":"Zebra Technologies",  
"device.company_contact":"123-555-1212",  
"device.location":"My Desk"}
```

➔ **Example 4** • To set several values at once:

```
{ }{"device.friendly_name":"XXQLJ120900310",  
"device.company_contact":"123-555-1212", "device.location":"My Desk"}
```

The response is:

```
{"device.friendly_name":"XXQLJ120900310",  
"device.company_contact":"123-555-1212", "device.location":"My Desk"}
```

➔ **Example 5** • If you do an `allconfig`, you can get the setting attributes for all settings as follows:

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

For the values used above it returns these entries:

```
"device.friendly_name":{"value":"XXQLJ120900310","type":"string","range":  
"0-17","clone":false,"archive":true,"access":"RW"},  
"device.company_contact":{"value":"123-555-1212","type":"string","range":  
"0-128","clone":true,"archive":true,"access":"RW"},  
"device.location":{"value":"my desk","type":"string","range":  
"0-128","clone":true,"archive":true,"access":"RW"},
```



**Note** • For the "allconfig" response, it will start with

```
  {"allconfig":{  
and end with  
  }}
```

## alerts.add

**Description** This command is used to configure the ZebraNet Alert System. It allows Zebra software to add new alerts without having to use the ZPL `^SX` command. This allows software to configure printers which do not have ZPL on them, and it provides the software with a single way in for configuring alerts. It also allows the software to configure alerts via local ports such as USB and serial.

The format is similar to the `^SX` command. It can delete the alert when both the set and clear flags are set to `FALSE`.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar`, `do`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | <p>This command instructs the printer to add the new alert with the configuration specified in the comma delimited list.</p> <p>The parameter is a comma delimited list of the following:</p> <ul style="list-style-type: none"> <li>• The alert condition. This can be any of the values returned from <a href="#">alerts.conditions on page 29</a>.</li> <li>• The alert destination type. This can be any of the value returned from <a href="#">alerts.destinations on page 31</a>.</li> <li>• On Set - Set to Y if the alert should be sent when the event is set</li> <li>• On Clear - Set to Y if the alert should be sent when the event is cleared</li> <li>• Destination address - applies to TCP,UDP,EMAIL,SNMP, SDK, and HTTP POST destination types</li> <li>• Port - Applies to TCP and UDP types</li> <li>• Quelling - When set to "Y" it prevents the alert from being sent. "N" is the default.</li> <li>• SGD Name - the name of the SGD command to be added. This is valid only when the alert condition is SGD_SET.</li> </ul> <p><i>Format:</i> ! U1 setvar "alerts.add" "[condition],[destination],[set],[clear],[destination_address],[port],[quelling],[SGD_name]"</p> <p><i>Values:</i> Defined via <a href="#">alerts.conditions on page 29</a>: PAPER OUT, RIBBON OUT, HEAD TOO HOT, HEAD COLD, HEAD OPEN, SUPPLY TOO HOT, RIBBON IN, REWIND, CUTTER JAM, MED, PRINTER PAUSED, PQ JOB COMPLETED, LABEL READY, HEAD ELEMENT BAD, BASIC RUNTIME, BASIC FORCED, POWER ON, CLEAN PRINthead, MEDIA LOW, RIBBON LOW, REPLACE HEAD, BATTERY LOW, RFID ERROR, ALL MESSAGES, COLD START, SGD SET</p> <p><i>Default:</i> NA</p> |
| <code>do</code>     | <p>This command has the same functionality as the <code>setvar</code>.</p> <p><i>Format:</i> ! U1 do "alerts.add" "[condition],[destination],[set],[clear],[destination_address],[port],[quelling],[SGD_name]"</p>   |



**Example** • This example shows a "Paper Out" alert sent via the serial port, with no destination address specified.

```
! U1 setvar "alerts.add" "PAPER OUT,SERIAL,Y,N,,0,,,"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## alerts.conditions

**Description** This command lists the available conditions that can be specified in the first parameter of the `alerts.add` SGD. See the [alerts.add on page 27](#) for information on the various parameters.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command retrieves the list of available alert conditions for the printer.</p> <p><i>Format:</i> ! U1 <code>getvar "alerts.conditions"</code></p> <p><i>Values:</i> PAPER OUT, RIBBON OUT, HEAD TOO HOT, HEAD COLD, HEAD OPEN, SUPPLY TOO HOT, RIBBON IN, REWIND, CUTTER JAM, MED, PRINTER PAUSED, PQ JOB COMPLETED, LABEL READY, HEAD ELEMENT BAD, BASIC RUNTIME, BASIC FORCED, POWER ON, CLEAN PRINthead, MEDIA LOW, RIBBON LOW, REPLACE HEAD, BATTERY LOW, RFID ERROR, ALL MESSAGES, COLD START, SGD SET</p> <p><i>Default:</i> ""</p> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# alerts.configured

**Description** This command creates a list of all the alerts that are configured on the printer. The alerts are delimited by the '|' character.



**Important** • Writing to this SGD will clear out the old alerts and setup the new ones.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar

| Commands | Details   |
|----------|---|
| setvar   | <p>This command creates the list of alerts configured on the printer.</p> <p><i>Format:</i> ! U1 setvar "alerts.configured" "&lt;a ' ' delimited list of configured alerts&gt;"</p> <p><i>Values:</i> A list of alerts to be setup on the printer. See alerts.add for the format of the individual alerts.</p> <p><i>Default:</i> "COLD START,SNMP,Y,N,255.255.255.255,162,N"</p> |
| getvar   | <p>This command retrieves the currently configured alerts on the printer.</p> <p><i>Format:</i> ! U1 getvar "alerts.configured"</p>   |



## Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# alerts.destinations

**Description** This command lists the available destinations that can be specified in the first parameter of the `alerts.add` SGD. See the [alerts.add](#) on page 27 for information on the various parameters.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>getvar</code> | <p>This command returns a comma-delimited list of available alert destinations.</p> <p><i>Format:</i> ! U1 <code>getvar "alerts.destinations"</code></p> <p><i>Values:</i> SERIAL, PARALLEL, E-MAIL, TCP, UDP, SNMP, USB, HTTP-POST, BLUETOOTH, SDK</p> <p><i>Default Value:</i> NA</p> |



### Note •

- For printer support, see *SGD Command Support* on page 1151.
- For details on SGD command structure, see *Command Structure* on page 636.

## alerts.http.authentication.add

**Description** This command allows the user to add a single server/username/password triplet into the list of authentication entries.

When the printer attempts to connect to the URL in the HTTP POST alert, the server may require HTTP authentication (such as digest, basic, DNS, etc.). There may be multiple authentication requests along the route to the destination (for example, a local server first requires HTTP authentication as well as on the remote server). For each HTTP authentication request received while attempting to connect, the printer will enumerate the authentication entries and attempt to satisfy the request with the username/password pair provided for the respective server. The server name in the entry is what determines which username/password pair should be used for which authentication request. Both DNS names and IP addresses are acceptable.

The server, username, and password are separated by a single space (not a tab or other white space character). The server name is the only required field. If no username is supplied, but a password is, there must be two spaces between the server and the password fields. If there is a username but no password, or simply just the servername, no space is required at the end of the entry.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, do

| Commands | Details   |
|----------|---|
| setvar   | <p>This command adds server/username/password triplet into the list of authentication entries.</p> <p><i>Format:</i><br/>! U1 setvar "alerts.http.authentication.add" "servername[ username][ password]"</p> <p><i>Values:</i> Maximum string of 2048 characters.</p> <p><i>Default:</i> NA</p> |
| do       | <p>This command has the same settings as the setvar.</p> <p><i>Format:</i><br/>! U1 do"alerts.http.authentication.add" "servername[ username][ password]"</p> <p><i>Values:</i> Maximum string of 2048 characters.</p> <p><i>Default:</i> NA</p>  |





**Example 1** • A username and a password is supplied:

```
! U1 setvar "alerts.http.authentication.add" "my.server.lan johndoe password"
```



**Example 2** • No password is supplied:

```
! U1 setvar "alerts.http.authentication.add" "my.server.lan johndoe"
```



**Example 3** • No username is supplied (note the double space):

```
! U1 setvar "alerts.http.authentication.add" "my.server.lan password"
```



**Example 4** • No username or password is supplied:

```
! U1 setvar "alerts.http.authentication.add" "my.server.lan"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

## alerts.http.authentication.entries

**Description** This command lists the server names added to the authentication entries list via `alerts.http.authentication.add`.

Only the server names will be shown; the username and passwords will not be shown. The server names are separated by a `\r\n` so that each shows up on its own line and is easier to read.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command lists the server names added to the authentication entry list.</p> <p><i>Format:</i> ! U1 <code>getvar "alerts.http.authentication.entries"</code></p> <p><i>Values:</i> A list of server names.</p> <p><i>Default:</i> NA</p> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

## alerts.http.authentication.remove

**Description** This command allows the user to remove a single server/username/password triplet from the list of authentication entries. To remove an entry, only the server name is supplied, and the entire entry will be removed. If an invalid entry is supplied no action is taken.

Note that the list of authentication triplets will be updated (and saved over a reset) but this SGD is just a command and doesn't have state. Therefore the persistent and restore defaults do not apply. The internal list that this command removes from, however, is persistent and defaultable (defaults to an empty list).

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, do

| Commands | Details   |
|----------|---|
| setvar   | This command adds server/username/password triplet into the list of authentication entries.<br><i>Format:</i> ! U1 setvar "alerts.http.authentication.remove" "servername"<br><i>Values:</i> Maximum string of 2048 characters.<br><i>Default:</i> NA |
| do       | This command has the same settings as the setvar.<br><i>Format:</i> ! U1 dp "alerts.http.authentication.remove" "servername"<br><i>Values:</i> Maximum string of 2048 characters.<br><i>Default:</i> NA   |



**Example** • A username and a password is supplied

```
! U1 setvar "alerts.http.authentication.remove" "my.server.lan"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

## alerts.http.logging.clear

**Description** This command clears the weblink alerts log entries. It does not disable logging. Setting this command to any value, including an empty string, will clear the weblink log entries.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, do

| Commands | Details   |
|----------|---|
| setvar   | <p>This command clears the weblink alerts log entries.</p> <p><i>Format:</i> ! U1 setvar "alerts.http.logging.clear" "value"</p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p> |
| do       | <p>This command clears the weblink alerts log entries.</p> <p><i>Format:</i> ! U1 do "alerts.http.logging.clear" "value"</p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p>     |



**Example** • This example clears the log entries with an empty string value.

```
! U1 setvar "alerts.http.logging.clear" ""
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

## alerts.http.logging.entries

**Description** This command returns the N number of entries in the http log, where N has a maximum value that is set by `alerts.http.logging.max_entries`.

The alerts http log is a collection of events related to sending HTTP POST messages. The log entries range anywhere from general status to errors that prevented a successful connection. Each log entry contains a timestamp for when it was logged by the system. The newest events will appear at the bottom of the list.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command returns the number of entries in the HTTP log.</p> <p><i>Format:</i> ! U1 <code>getvar "alerts.http.logging.entries"</code></p> <p><i>Values:</i> NA</p> <p><i>Default:</i> NA</p> |



**Example** • This example shows the result from `alerts.http.logging.entries`:

```
[01-03-2013 12:48:59.964] [http] Connected to 10.3.4.58 (10.3.4.58) port 80
[01-03-2013 12:48:59.978] [http] HTTP/1.1 100 Continue
[01-03-2013 12:49:01.999] [http] Closing connection
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

# alerts.http.logging.max\_entries

**Description** This command specifies the maximum number of individual log entries that will be stored in the `alert.http.logging.entries` command.



**Important** • Changes to this command are immediate and may result in some log entries being lost. If there are N log entries currently in the log, the user sets the `max_entries` to M, where M is less than N, the oldest (N-M) log entries will be removed.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar, do`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | This command sets the maximum number of log entries that will be stored.<br><i>Format:</i> <code>! U1 getvar "alerts.http.logging.max_entries" "value"</code><br><i>Values:</i> 0 - 10000<br><b>Note</b> • Setting the value to 0 disables logging.<br><i>Default:</i> "0" |
| <code>getvar</code> | This command returns the setting for the maximum number of log entries that will be stored.<br><i>Format:</i> <code>! U1 getvar "alerts.http.logging.max_entries"</code>   |
| <code>do</code>     | This command sets the maximum number of log entries that will be stored.<br><i>Format:</i> <code>! U1 do "alerts.http.logging.max_entries" "value"</code><br><i>Values:</i> 0 - 10000<br><b>Note</b> • Setting the value to 0 disables logging.<br><i>Default:</i> "0"     |



**Example** • In this example, `alert.http.logging.max_entries` is then set to 2.

```
[01-03-2013 12:48:59.964] [http] Connected to 10.3.4.58 (10.3.4.58) port 80
[01-03-2013 12:48:59.978] [http] HTTP/1.1 100 Continue
[01-03-2013 12:49:01.999] [http] Closing connection
```

When it is set to 2

```
[01-03-2013 12:48:59.978] [http] HTTP/1.1 100 Continue
[01-03-2013 12:49:01.999] [http] Closing connection
```



## Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.

## alerts.http.proxy

**Description** This command assigns the URL of the proxy for any HTTP POST alerts. The proxy server protocol, port, domain, username, and password are all encoded into the URL via the format outlined in RFC2396 (<http://www.ietf.org/rfc/rfc2396.txt>).

The username and password must avoid the invalid characters listed in RFC2396 (e.g. ':', '@', '/', etc). If an invalid character must be used it needs to be escaped using '%' as described in RFC2396.

When the setting is changed, the next HTTP POST alert will use the new value.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar, do

| Commands | Details  |
|----------|--|
| setvar   | <p>This command assigns the proxy URL for HTTP POST alerts.</p> <p><i>Format:</i> ! U1 setvar "alerts.http.proxy"<br/>"http://username:password@mydomain.com:3128/"</p> <p><i>Values:</i> Any valid URL up to 2048 characters<br/>URL format expected: <code>http://[user:pass@]domain[:port]/[path]</code></p> <p><i>Default:</i></p> <ul style="list-style-type: none"> <li>• The user:pass, port, and path are all optional.</li> <li>• The default port is 1080.</li> <li>• The default is to omit the username and password.</li> </ul> |
| getvar   | <p>This command retrieves the proxy URL for HTTP POST alerts.</p> <p><i>Format:</i> ! U1 getvar "alerts.http.proxy"</p>  |
| do       | <p>This command assigns the proxy URL for HTTP POST alerts.</p> <p><i>Format:</i> ! U1 do "alerts.http.proxy"<br/>"http://username:password@mydomain.com:3128/"</p> <p><i>Values:</i> Any valid URL up to 2048 characters<br/>URL format expected: <code>http://[user:pass@]domain[:port]/[path]</code></p> <p><i>Default:</i></p> <ul style="list-style-type: none"> <li>• The user:pass, port, and path are all optional.</li> <li>• The default port is 1080.</li> <li>• The default is to omit the username and password.</li> </ul>     |



**Example** • Examples of how to connect to various proxy servers:

`http://username:password@mydomain.com:3128/`  
`http://mydomain.com/`



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.
- For details on HTTP POST, see *HTTP POST Alerts on page 17*.



# alerts.tracked\_settings.log\_tracked

**Description** This command creates a comma-delimited list of settings for which sets should be logged.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar, do

| Commands | Details   |
|----------|---|
| setvar   | <p>This command sets the list of settings for which sets should be logged.</p> <p><i>Format:</i> ! U1 setvar "alerts.tracked_settings.log_tracked" "settings.name1,settings.name2..."</p> <p><i>Values:</i> Settings with commas between names.</p> <p><i>Default:</i> ""</p> |
| getvar   | <p>This command returns a comma-delimited lists of settings being logged.</p> <p><i>Format:</i> ! U1 getvar "alerts.tracked_settings.log_tracked"</p>   |
| do       | <p>This command sets the list of settings for which sets should be logged.</p> <p><i>Format:</i> ! U1 do "alerts.tracked_settings.log_tracked" "settings.name1,settings.name2..."</p> <p><i>Values:</i> Settings with commas between names.</p> <p><i>Default:</i> ""</p>     |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## alerts.tracked\_settings.clear\_log

**Description** This command clears the `alerts.tracked_settings.log`. Setting this command to any value, including an empty string, will clear the `tracked_sgds` log entries.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar`, `do`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command clears the <code>tracked_sgds</code> log entries.</p> <p><i>Format:</i> <code>! U1 setvar "alerts.tracked_settings.clear_log" "value"</code></p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p> |
| <code>do</code>     | <p>This command clears the <code>tracked_sgds</code> log entries.</p> <p><i>Format:</i> <code>! U1 do "alerts.tracked_settings.clear_log" "value"</code></p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p>     |



**Example** • This example clears the log entries with an empty string value.

```
! U1 setvar "alerts.tracked_settings.clear_log" ""
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## alerts.tracked\_sgds.log

**Description** This command reports the log of the settings listed in `alerts.tracked_settings.log_tracked`. The log entries will be fully JSON compliant.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | This command retrieves the current log.<br><i>Format:</i> <code>! U1 getvar "alerts.tracked_sgds.log"</code> |



**Example** • Sending `! U1 getvar "alerts.tracked_settings.log"` returns:

```
:[{"settingsName":"newValue","timestamp"
:"06-24-2012 19:51:28.641"}]" for 1 entry or
"[{"settingsName":"newValue","timestamp"
:"06-24-2012
19:51:28.641"},\r\n{"settingsName2":"newValue2","timestamp":"06-24-2012
19:51:30.641"}]" for 2 entries.
```

**Note** • When the log is empty, the result will be:

```
""
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# alerts.tracked\_sgds.max\_log\_entries

**Description** This command sets the maximum number of entries to be shown in `alerts.tracked_settings.log`.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar, do`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command sets the maximum number of alert log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 setvar "alerts.tracked_sgds.max_log_entries" "value"</code></p> <p><i>Values:</i> 0 - 10000</p> <p><b>Note</b> • Setting the value to 0 disables logging.</p> <p><i>Default:</i> "100"</p> |
| <code>getvar</code> | <p>This command returns the setting for the maximum number of alert log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 getvar "alerts.tracked_sgds.max_log_entries"</code></p>   |
| <code>do</code>     | <p>This command sets the maximum number of alert log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 setvar "alerts.tracked_sgds.max_log_entries" "value"</code></p> <p><i>Values:</i> 0 - 10000</p> <p><b>Note</b> • Setting the value to 0 disables logging.</p> <p><i>Default:</i> "100"</p> |



**Example** • This example sets the maximum log entries to 50.

```
! U1 setvar "alerts.tracked_sgds.max_log_entries" "50"
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## alerts.tracked\_sgds.zbi\_notified

**Description** This command provides a comma-delimited list of settings for which ZBI should be notified when the value is set.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar,getvar,do

| Commands | Details  |
|----------|--|
| setvar   | <p>This command sets the list of the settings for which ZBI will be notified when the setting is set.</p> <p><i>Format:</i> ! U1 setvar "alerts.tracked_settings.zbi_notified" "settings.name1,settings.name2,etc."</p> <p><i>Values:</i> A comma delimited list of settings names</p> <p><i>Default:</i> ""</p> |
| getvar   | <p>This command retrieves the list of the settings for which ZBI will be notified when the value is set.</p> <p><i>Format:</i> U1 getvar "alerts.tracked_settings.zbi_notified"</p>  |
| do       | <p>This command sets the list of the settings for which ZBI will be notified when the value is set.</p> <p><i>Format:</i> ! U1 do "alerts.tracked_settings.zbi_notified" "settings.name1,settings.name2,etc."</p> <p><i>Values:</i> A comma delimited list of settings names</p> <p><i>Default:</i> ""</p>       |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## appl.link\_os\_version

**Description** This command lists the version of the Link-OS™ feature set that is supported by the printer.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | This command retrieves the Link-OS™ version of the printer.<br><i>Format: ! U1 getvar "appl.link_os_version"</i> |



**Example** • In this example, the `getvar` command returns version 1.0 of Link-OS™.

```
! U1 getvar "appl.link_os_version"
returns
1.0
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## capture.channel1.count

**Description** This command indicates the number of times that `capture.channel1.delimiter` was seen on the port specified in `capture.channel1.port`. Additionally, it indicates how many times `capture.channel1.data.raw` has been updated with user data as well as the number of times we reached the `capture.channel1.max_length`.

This will be shown in the [HZA](#) response under the capture data section.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>getvar</code> | <p>This command returns the number of times that <code>capture.channel1.delimiter</code> was seen on the port specified in <code>capture.channel1.port</code> as well as the number of times we reached the <code>capture.channel1.max_length</code>.</p> <p><i>Format: ! U1 getvar "capture.count"</i></p> |



#### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## capture.channel1.data.mime

**Description** This command provides a view to the data captured on the port specified by `capture.channel1.port` in a mime/base64 encoded format.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>getvar</code> | <p>This command retrieves the data captured on the port specified by <code>capture.channel1.port</code>.</p> <p><i>Format:</i> ! U1 <code>getvar "capture.channel1.data.mime"</code></p> <p><i>Result:</i> Data in mime-encoded format.</p> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.



## capture.channel1.data.raw

**Description** This command retrieves the user data captured off of the port specified in `capture.channel1.port`.

Any binary zeros in the `capture.data` stream will be replaced with the escaped representation of NULL ("`\000`"). The delimiter data is not stored as part of the captured data.

This will be shown in the [HZA](#) output within capture data section.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command retrieves the user data captured off of the port specified in <code>capture.channel1.port</code>.</p> <p><i>Format:</i> ! U1 <code>getvar "capture.channel1.data.raw"</code></p> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# capture.channel1.delimiter


**Description** This command stores the delimiter used to partition data received on the port specified by `capture.channel1.port` and stored in `capture.channel1.data.raw` and `capture.channel1.data.mime`.

This will be reported in the data capture section of the HZA response.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar

| Commands | Details   |
|----------|---|
| setvar   | <p>This command sets the delimiter used to partition data received on the <code>capture.channel1.port</code>.</p> <p><i>Format:</i> ! U1 setvar "capture.channel1.delimiter" "delimiter"</p> <p><i>Values:</i> Any character set up to a maximum of 64 characters in length.</p> <p> <b>Note</b> • Binary data can be used in the delimiter. To do this enter a '\ ' and then the 3 digit octal value of the character. "\\ " = '\ ' in some tools, so to get \002 you might enter "\\002". Escaped octal characters count as a single character and not 4 (e.g. a delimiter of "\001\000\002" is 3 characters, not 12)</p> <p><i>Default:</i> "\012"</p> |
| getvar   | <p>This command retrieves the delimiter.</p> <p><i>Format:</i> ! U1 getvar "capture.channel1.delimiter"</p>   |



**Example** • Binary data can be used in the delimiter. To do this enter a '\ ' and then the 3 digit octal value of the character. Note: "\\ " = '\ ' in some tools, so to get \002 you might enter "\\002".

"\000" = NULL (single character)

"end\015\012\000" = 'e'+ 'n'+ 'd'+ '\r'+ '\n'+ NULL (total of 6 characters)



## Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## capture.channel1.max\_length

**Description** This command sets a length indicating when to copy captured data to the data SGD if the delimiter has not been seen yet.

If the delimiter and the max\_length are reached at the same time, the delimiter will not be part of the captured data. If only part of the delimiter has been received, then the part of the delimiter we have received, will be part of the capture data.

When the max\_length is changed, any data currently in the buffer will be thrown away, and the new value of max\_length will be used.

The Capture Port shall be defaulted to 1000 bytes by any mechanism (including ^JUF, ^JUN, ^JUA, and device.restore\_defaults).

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar

| Commands | Details   |
|----------|---|
| setvar   | This command instructs the printer to set a default data capture length.<br><i>Format:</i> ! U1 setvar "capture.channel1.max_length" "value"<br><i>Values:</i> 1-3000<br><i>Default:</i> "1000" |
| getvar   | This command retrieves the default data capture length.<br><i>Format:</i> ! U1 getvar "capture.channel1.max_length"   |



#### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## capture.channel1.port

**Description** This command determines the port that should be monitored for user data. This allows the user to attach an external device, such as a keyboard or barcode scanner, and have input captured into the `capture.channel1.data.raw` command. Once the data is in the SGD they can use it as they would any other SGD (this includes functionality that allows users to be sent an alert when an SGD value changes).

The data received on the specified port will be read until the value in `capture.channel1.delimiter` is seen, at which point the data received until (but not including) the delimiter will be stored in `capture.channel1.data.raw`.

For the port specified in `capture.channel1.port`, no data will be sent to any of the parsers on that port. All data received is assumed to be user input that is to be placed in `capture.channel1.data.raw`. To disable the data capture functionality, set `capture.channel1.port` to "off"

- The delimiter will not be stored in `capture.channel1.data.raw`.
- The port will be shown in the data capture portion of the HZA response.
- The capture port shall be defaulted to "off" by any mechanism (including ^JUF, ^JUA, and `device.restore_defaults`).

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command sets the port to be monitored for user data.</p> <p><i>Format:</i> ! U1 setvar "capture.channel1.port" "value"</p> <p><i>Values:</i> off, serial, usb, bt, usb_host</p> <ul style="list-style-type: none"> <li>• <code>off</code> - no data is stored in <code>capture.channel1.data.raw</code> and all data is sent to the pasers - normal operation</li> <li>• <code>serial</code> - Data is read off the serial port. No data sent to the parsers on this port.</li> <li>• <code>usb</code> - Data is read off the usb port. No data sent to the parsers on this port.</li> <li>• <code>bt</code> - Data is read off the Bluetooth® port. No data sent to the parsers on this port.</li> <li>• <code>usb_host</code> - not yet supported. reserved for when usb host is implemented.</li> </ul> <p><i>Default:</i> "off"</p> |
| <code>getvar</code> | <p>This command retrieves the printer's current port being monitored for user data.</p> <p><i>Format:</i> ! U1 getvar "capture.channel1.port"</p>   |



**Example** • This example sets the command value to "off", preventing it from capturing data.

```
! U1 setvar "capture.channel11.port" "off"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## device.user\_vars.create

**Description** This command creates a user variable with the specified name, type, range, and default value.

The root SGD location for user variables is "device.user\_vars".

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar

| Commands | Details   |
|----------|---|
| setvar   | <p>This command instructs the printer to create a user variable with the specified parameters.</p> <p><i>Format:</i> ! U1 setvar "device.user_vars.create" "name:type:range:defaultValue"</p> <p><i>Values:</i></p> <ul style="list-style-type: none"> <li>• <b>name</b> = the name of the SGD to appear in device.user_vars. The name can be anything from 1 to 64 printable ASCII characters. Any '.' within the name will be replaced with '_'. (e.g. "john.doe" will be changed to "john_doe". The name must be unique in the device.user_vars branch or it will not be created. The name will be converted to lower case.</li> <li>• <b>type</b> = one of STRING, INTEGER, DOUBLE, CHOICES, UPDOWNCHOICES, UPDOWNINTEGER, UPDOWNDOUBLE. The type must be one of these types or the variable will not be created. <ul style="list-style-type: none"> <li>• STRING - For strings the range indicates the min/max length of the data that can be stored. If the range is left blank, the default range is a string length of 0-1024. There is no maximum string length, however, if large data is placed into the variables the user should be aware that system memory and performance will be affected. Strings larger than available system memory will not be stored. Values should attempt to stay around, or below, 5K.</li> <li>• INTEGER/UPDOWNINTEGER - For integers the range can be any number expressed by a 32-bit integer, signed or unsigned. If the range is left blank then a range of -32768 to 32767 will be used.</li> <li>• DOUBLE/UPDOWNDOUBLE - A double can be any value within the range of +/-1.7e308. If the range is left blank then a range of -32768.0 to 32767.0 will be used.</li> <li>• CHOICES/UPDOWNCHOICES - Choices must be specified in a comma delimited list. The range cannot be blank if the type is CHOICES or UPDOWNCHOICES.</li> </ul> </li> <li>• <b>range</b> = Expressed as x-y. The range of a variable depends upon the type. Some types will create default ranges, while others will fail to be created if the range is invalid or not specified.</li> <li>• <b>default</b> = the default value for the variable. The value must fall within the range specified or the variable will not be created. If the type is INTEGER, UPDOWNINTEGER, DOUBLE, UPDOWNDOUBLE the default value will be 0 if not specified. For STRING the default value will be an empty string if it is not specified. CHOICES and UPDOWNCHOICES must have a default value and it must be one of the choices within the specified range.</li> <li>• All four parts of the creation string must be present (some can be empty) meaning that there must be three delimiter characters (':') present. There is no error shown or indicated otherwise when the variable is not created for some reason. If the variable is not created one of the rules listed above has been violated.</li> <li>• Any user variables will be deleted from the device.user_vars branch on a power cycle (they won't be recreated on the next power up).</li> <li>• Defaulting the user_vars branch will restore the consumers back to their defaulted values and will not remove them from the user_vars branch.</li> </ul> <p><i>Default:</i> NA</p> |



**Example •** To create a user variable named `userVar1` that is an integer with a minimum of 1, a maximum of 10, and a default/initial value of 5, issue this command:

```
! U1 setvar "device.user_vars.create" "userVar1:INTEGER:1-10:5"
```

After issuing the above “create” command the `device.user_vars.userVar1` SGD will be present in an ALLCV response.

After issuing the above “create” command the `device.user_vars.userVar1` SGD may be set via:

```
! U1 setvar "device.user_vars.userVar1" "2"
```

After issuing the above “create” command the “`device.user_vars.userVar1`” SGD may be retrieved via:

```
! U1 getvar "device.user_vars.userVar1"
```



**Note •**

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## device.user\_vars.set\_range

**Description** This command compliments the `device.user_vars.create` command, allowing a user to change the range of a user-created SGD variable. It has a similar syntax to `device.user_vars.create` with the exception that no default is specified.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands | Details  |
|----------|--|
| setvar   | <p>This command allows the user to change the range of a user-created variable.</p> <p><i>Format:</i> <code>! U1 setvar "device.user_vars.set_range" "name:type:range"</code></p> <p><i>Values:</i></p> <ul style="list-style-type: none"> <li>• <code>name</code> = the name of the SGD to modify</li> <li>• <code>type</code> = Must be the same type for 'name' as when it was created</li> <li>• <code>range</code> = x-y (for all but UPDOWNCHOICES and CHOICES) or a,b,c,d (for CHOICES and UPDOWNCHOICES)</li> <li>• If no range is specified then it will delete whatever range is currently specified.</li> </ul> <p><i>Default:</i> NA</p> |



**Example** • This example modifies my\_var to:

```
device.user_vars.my_var : b , Choices: a,b,c,d,e
```

```
! U1 setvar "device.user_vars.set_range" "my_var:CHOICES:a,b,c,d,e"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.



# internal\_wired.ip.port\_json\_config

**Description** This command determines the TCP port number to listen on for JSON configuration packets.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar

| Commands | Details   |
|----------|---|
| setvar   | <p>This command sets the TCP port number to listen on for JSON configuration packets.</p> <p><i>Format:</i> ! U1 setvar "internal_wired.ip.port_json_config" "value"</p> <p><i>Values:</i></p> <ul style="list-style-type: none"> <li>• 0 = disable the port</li> <li>• 1-65535 = Port number to listen on.</li> </ul> <p><b>Note</b> • Ports that are already in use or the standard network ports are invalid values.</p> <p><i>Default:</i> "9200"</p> |
| getvar   | <p>This command retrieves the port number.</p> <p><i>Format:</i> ! U1 getvar "internal_wired.ip.port_json_config"</p>   |



**Example** • This example sets the port value to listen on as 1234.

```
! U1 setvar "internal_wired.ip.port_json_config" "1234"
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## ip.port\_json\_config

**Description** This command determines the TCP port number to listen on for JSON configuration packets.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, getvar

| Commands | Details  |
|----------|--|
| setvar   | <p>This command sets the TCP port number to listen on for JSON configuration packets.</p> <p><i>Format:</i> ! U1 setvar "ip.port_json_config" "value"</p> <p><i>Values:</i> Any valid port number except ports already in use or the standard network ports.</p> <ul style="list-style-type: none"> <li>• 0 = disable the port</li> <li>• 1-65535 = Port number to listen on.</li> </ul> <p><i>Default:</i> "9200"</p> |
| getvar   | <p>This command retrieves the port number.</p> <p><i>Format:</i> ! U1 getvar "ip.port_json_config"</p>   |



**Example** • This example sets the port value to listen on as 12444.

```
! U1 setvar "ip.port_json_config" "12444"
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# weblink.enable

**Description** This command indicates if one or more of the weblink connections are active.

If there is more than one connection under the weblink branch (e.g. `weblink.ip.conn1`) and if any of the `.location` values are set, then this SGD will be set to "on". If all connections are disabled (all connection `.location` values set to ""), then this value will be set to "off".

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command indicates if one or more of the weblink connections are active.</p> <p><i>Format:</i> ! U1 <code>getvar "weblink.enable"</code></p> <p><i>Result:</i></p> <ul style="list-style-type: none"> <li>• "yes" if any of the <code>.location</code> values are set</li> <li>• "off" if all connectoins are disabled</li> </ul> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].authentication.add

**Description** This command allows the user to add a single server/username/password triplet into the list of authentication entries.

When the printer attempts to connect to the server (url specified in `weblink.ip.conn[1|2].location`) the server may require HTTP authentication (e.g. digest, basic, DNS, etc.). There may be multiple authentication requests along the route to the destination (e.g. a local server first requires HTTP authentication as well as on the remote server).

For each HTTP authentication request received while attempting to connect, the printer will enumerate the authentication entries and attempt to satisfy the request with the username/password pair provided for the respective server. The server name in the entry is what determines which username/password pair should be used for which authentication request. Both DNS names and IP addresses are acceptable. The server, username, and password are separated by a single space (not a tab or other white space character). The server name is the only required field. If no username is supplied, but a password is, there must be two spaces between the server and the password fields. If there is a username but no password, or simply just the servername, no space is required at the end of the entry.

If the command is changed when the connection is enabled (`weblink.ip.conn[1|2].enable`), it will not take effect until the connection is disabled and then re-enabled.



**Important** • This setting only be changed when `weblink.ip.conn1.enable` is set to "off".

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | <p>This command adds a single server/username/password triplet to the list of authentication entries.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 setvar "weblink.ip.conn1.authentication.add" "servername[ username][ password]"</code></li> <li>• <code>! U1 setvar "weblink.ip.conn2.authentication.add" "servername[ username][ password]"</code></li> </ul> <p><i>Values:</i> <code>servername [username][ password]</code> has a maximum length of string is 2048 characters</p> <p><i>Default:</i> NA</p> |



**Example 1** • In this example, a username and a password is supplied:

```
! U1 setvar "weblink.ip.conn1.authentication.add" "my.server.lan johndoe  
password"
```

**Example 2** • In this example, no password is supplied

```
! U1 setvar "weblink.ip.conn1.authentication.add" "my.server.lan johndoe"
```

**Example 3** • In this example, no username is supplied (note the double space)

```
! U1 setvar "weblink.ip.conn1.authentication.add" "my.server.lan password"
```

**Example 4** • In this example, no username or password is supplied

```
! U1 setvar "weblink.ip.conn1.authentication.add" "my.server.lan"
```



**Note** •

- For printer support, see *SGD Command Support* on page 1151.
- For details on SGD command structure, see *Command Structure* on page 636.

## weblink.ip.conn[1|2].authentication.entries

**Description** This command lists the server names added to the authentication entries list.

Only the server names will be shown. The username and passwords will not be shown. The server names are separated by a `\r\n` so that each shows up on its own line.

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>getvar</code> | <p>This command lists the server names for the specified connection.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 getvar "weblink.ip.conn1.authentication.entries"</code></li> <li>• <code>! U1 getvar "weblink.ip.conn2.authentication.entries"</code></li> </ul> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].authentication.remove

**Description** This command allows the user to remove a single server/username/password triplet from the list of authentication entries.

To remove an entry only the server name is supplied. If an invalid entry is supplied no action is taken. If the SGD is changed when the connection is enabled (`weblink.ip.conn[1|2].enable`), it will not take effect until the connection is disabled and then re-enabled. It is therefore recommended that this setting only be changed when `weblink.ip.conn[1|2].enable` is set to "off".

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands | Details  |
|----------|--|
| setvar   | <p>This command removes a single server/username/password triplet from the list of authentication entries.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 setvar "weblink.ip.conn1.authentication.remove" "servername"</code></li> <li>• <code>! U1 setvar "weblink.ip.conn2.authentication.remove" "servername"</code></li> </ul> <p><i>Values:</i> <code>servername</code> has a maximum length of string is 2048 characters.</p> <p><i>Default:</i> NA</p> |



**Example** • In this example, a username and a password is supplied

```
! U1 setvar "weblink.ip.conn1.authentication.remove" "my.server.lan"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].location

**Description** This command assigns the URL of the server for this connection. The URL must follow the URL rules for the HTTP[S] protocol outlined in RFC2396 (<http://www.ietf.org/rfc/rfc2396.txt>).

The setting will not take effect until the printer is reset. Changing this setting will set `weblink.printer_reset_required` to "yes".

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | <p>This command sets the server URL for the specified connection.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 getvar "weblink.ip.conn1.location" "url"</code></li> <li>• <code>! U1 getvar "weblink.ip.conn2.location" "url"</code></li> </ul> <p><i>Values:</i> any HTTPS URL up to 2048 characters</p> <p><i>Default:</i> ""</p> |
| <code>getvar</code> | <p>This command returns the server URL currently assigned to the connection.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 getvar "weblink.ip.conn1.location"</code></li> <li>• <code>! U1 getvar "weblink.ip.conn2.location"</code></li> </ul>  |



### Example •

```
! U1 setvar "weblink.ip.conn2.location"
"https://my.linkos.server.com:8080/link/os"
```



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.



## weblink.ip.conn[1|2].maximum\_simultaneous\_connections

**Description** This command indicates the maximum number of simultaneous connections that can be initiated by the printer.

Via the main connection (the original connection initiated by the printer to the remote server), the remote server can request that additional connections from the printer be initiated (e.g. a connection that supports only JSON SGDs, one that behaves similar to the RAW TCP port).

The server is free to request as many as it thinks it needs, but the printer will prevent more than N number of connections, where N is the value of this command.

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`, `setvar`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | <p>This command sets the maximum number of connections.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 do "weblink.ip.conn1.maximum_simultaneous_connections" "value"</code></li> <li>• <code>! U1 do "weblink.ip.conn2.maximum_simultaneous_connections" "value"</code></li> </ul> <p><i>Values:</i> Any interger from 1-100</p> <p><i>Default:</i> "10"</p> |
| <code>getvar</code> | <p>This command retrieves the maximum set number of connections.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 do "weblink.ip.conn1.maximum_simultaneous_connections"</code></li> <li>• <code>! U1 do "weblink.ip.conn2.maximum_simultaneous_connections"</code></li> </ul>  |



**Example** • This example sets the conn1 maximum connections to 3.

```
! U1 setvar "weblink.ip.conn1.maximum_simultaneous_connections" "3"
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1 | 2].proxy

**Description** This command assigns the URL of the proxy for the connection.

The proxy server protocol, port, domain, username, and password are all encoded into the URL via the format outlined in RFC2396 (<http://www.ietf.org/rfc/rfc2396.txt>).

The username and password must avoid the invalid characters listed in RFC2396 (e.g. ':', '@', '/', etc). If an invalid character must be used it needs to be escaped using '%' as described in RFC2396.

^JUF, ^JUS, ^JUN, ^JUA, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar, do`

| Commands | Details  |
|----------|--|
| setvar   | <p>This command assigns the URL of the connection proxy.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 setvar "weblink.ip.conn1.proxy" "url"</li> <li>• ! U1 setvar "weblink.ip.conn2.proxy" "url"</li> </ul> <p><i>Values:</i></p> <ul style="list-style-type: none"> <li>• Any valid URL up to 2048 characters</li> <li>• Expected URL format: [http https]://[user:pass@]domain[:port]/[path]</li> <li>• The URL will need to be built according to the server/proxy environment the printer is running within.</li> </ul> <p><i>Default: ""</i></p> <ul style="list-style-type: none"> <li>• The user:pass, port, and path are all optional.</li> <li>• The default scheme must be either HTTPS or HTTP. The default is HTTP.</li> <li>• The default port is 1080.</li> <li>• The default is to omit the username and password.</li> </ul> |
| getvar   | <p>This command retrieves the URL of the connection proxy.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 getvar "weblink.ip.conn1.proxy"</li> <li>• ! U1 getvar "weblink.ip.conn2.proxy"</li> </ul>  |

| Commands | Details  |
|----------|--|
| do       | <p>This command assigns the URL of the connection proxy.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 do "weblink.ip.conn1.proxy" "url"</li> <li>• ! U1 do "weblink.ip.conn2.proxy" "url"</li> </ul> <p><i>Values:</i></p> <ul style="list-style-type: none"> <li>• Any valid URL up to 2048 characters</li> <li>• Expected URL format: [scheme://][user:pass@]domain[:port]/[path]</li> <li>• The URL will need to be built according to the server/proxy environment the printer is running within.</li> </ul> <p><i>Default: ""</i></p> <ul style="list-style-type: none"> <li>• The scheme, user:pass, port, and path are all optional.</li> <li>• The default scheme is http.</li> <li>• The default port is 1080.</li> <li>• The default is to omit the username and password.</li> </ul> |



**Example** • Examples of how to connect to various proxy servers:

```
http://username:password@mydomain.com:3128/
http://mydomain.com/
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# weblink.ip.conn[1|2].retry\_interval

**Description** This command sets the number of seconds between attempts to connect to the server URL provided in `weblink.ip.conn1.location`. If an attempt is unsuccessful or the connection is lost, the printer will wait '`retry_interval`' seconds before attempting to connect again.

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any effect on this setting.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command sets the number of seconds to wait before attempting to reconnect to the server.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 setvar "weblink.ip.conn1.retry_interval" "value"</li> <li>• ! U1 setvar "weblink.ip.conn2.retry_interval" "value"</li> </ul> <p><i>Values:</i> 1 - 600</p> <p><i>Default:</i> "10"</p> |
| <code>getvar</code> | <p>This command returns the number of seconds to wait between connection attempts.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 getvar "weblink.ip.conn1.retry_interval"</li> <li>• ! U1 getvar "weblink.ip.conn2.retry_interval"</li> </ul>   |
| <code>do</code>     | <p>This command sets the number of seconds to wait before attempting to reconnect to the server.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 do "weblink.ip.conn1.retry_interval" "value"</li> <li>• ! U1 do "weblink.ip.conn2.retry_interval" "value"</li> </ul> <p><i>Values:</i> 1 - 600</p> <p><i>Default:</i> "10"</p>         |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].test.location

**Description** This command holds the URL for testing a connection to the internet. This is meant to assist users in debugging their printer's connection to remote servers when there are issues with the main weblink connection (conn1 or conn2).

The URL must follow the URL rules for the HTTP[S] protocol outlined in RFC2396 (<http://www.ietf.org/rfc/rfc2396.txt>).

^JUF, ^JUS, ^JUN, ^JUA, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar,getvar,do

| Commands | Details   |
|----------|---|
| setvar   | <p>This command sets the URL to hold for testing a connection.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 setvar "weblink.ip.conn1.test.location" "url"</li> <li>• ! U1 setvar "weblink.ip.conn2.test.location" "url"</li> </ul> <p><i>Values:</i> Any HTTPS URL up to 2048 characters</p> <p><i>Default:</i> "http://www.zebra.com/apps/linktest"</p> |
| getvar   | <p>This command retrieves the printer's test connection URL.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 getvar "weblink.ip.conn1.test.location"</li> <li>• ! U1 getvar "weblink.ip.conn2.test.location"</li> </ul>   |
| do       | <p>This command sets the URL to hold for testing a connection.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 do "weblink.ip.conn1.test.location" "url"</li> <li>• ! U1 do "weblink.ip.conn2.test.location" "url"</li> </ul> <p><i>Values:</i> Any HTTPS URL up to 2048 characters</p> <p><i>Default:</i> "http://www.zebra.com/apps/linktest"</p>         |



**Example •** The test connection can assist the user in several ways/scenarios:

1. If the `test.test_on` value is set to "failure", any time the main weblink (`conn[1|2].location`) connection fails to connect then the `test.location` URL will be used. In this situation, an attempt will be made to contact the remote URL in `test.location`, using authentication and proxy configuration that is specified by the main connection.
2. If the `test.test_on` value is set to "interval" an attempt will be made to contact the remote URL in `test.location` every `test.retry_interval` seconds, using authentication and proxy configuration that is specified by the main connection.
3. If the `test.test_on` value is set to "both", then scenario 1 and 2 will both occur. This is useful for users who will use an HTTP connection to move through their firewall - and thereafter frequently refresh the connection to indicate to their firewall that there is still activity for the purpose of keeping the connection alive.



**Note •**

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].test.retry\_interval

**Description** This command determines how often, in seconds, a connection to the test.location URL should be attempted. This setting is only applicable when the `test.test_on` SGD is set to "interval" or "both".

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setevar`, `getvar`, `do`

| Commands | Details   |
|----------|---|
| setvar   | <p>This command sets the interval for how often a connection to the test.location URL should be attempted.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 setvar "weblink.ip.conn1.test.retry_interval" "value"</li> <li>• ! U1 setvar "weblink.ip.conn2.test.retry_interval" "value"</li> </ul> <p><i>Values:</i> 0-1800 (in seconds, providing 0 second - 30 minute interval)</p> <p><i>Default:</i> "900"</p> |
| getvar   | <p>This command retrieves the retry interval.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 getvar "weblink.ip.conn1.test.retry_interval"</li> <li>• ! U1 getvar "weblink.ip.conn2.test.retry_interval"</li> </ul>  |
| do       | <p>This command sets the interval for how often a connection to the test.location URL should be attempted.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• ! U1 do "weblink.ip.conn1.test.retry_interval" "value"</li> <li>• ! U1 do "weblink.ip.conn2.test.retry_interval" "value"</li> </ul> <p><i>Values:</i> 0-1800</p> <p><i>Default:</i> "900"</p>   |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.ip.conn[1|2].test.test\_on

**Description** This command determines when the test connection should be attempted. This assists in debugging the printer's connection to remote servers when there are issues with the main weblink connection (conn1 or conn2).

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>setvar</code> | <p>This command indicate when the test connection should be attempted.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 setvar "weblink.ip.conn1.test.test_on" "value"</code></li> <li>• <code>! U1 setvar "weblink.ip.conn2.test.test_on" "value"</code></li> </ul> <p><i>Values:</i> <code>off</code>, <code>failure</code>, <code>interval</code>, <code>both</code></p> <p><i>Default:</i> <code>"failure"</code></p> |
| <code>getvar</code> | <p>This command retrieves the test connection setting.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 setvar "weblink.ip.conn1.test.test_on"</code></li> <li>• <code>! U1 setvar "weblink.ip.conn2.test.test_on"</code></li> </ul>  |
| <code>do</code>     | <p>This command sets when the test connection should be attempted.</p> <p><i>Format:</i></p> <ul style="list-style-type: none"> <li>• <code>! U1 do "weblink.ip.conn1.test.test_on" "value"</code></li> <li>• <code>! U1 do "weblink.ip.conn2.test.test_on" "value"</code></li> </ul> <p><i>Values:</i> <code>off</code>, <code>failure</code>, <code>interval</code>, <code>both</code></p> <p><i>Default:</i> <code>"failure"</code></p>             |





**Example •** The test connection can assist the user in several ways/scenarios:

1. If the `test.test_on` value is set to "failure", any time the main weblink (`conn[1|2].location`) connection fails to connect then the `test.location` URL will be used. An attempt will be made to contact the remote URL in `test.location`, using authentication and proxy configuration that is specified by the main connection.
2. If the `test.test_on` value is set to "interval" an attempt will be made to contact the remote URL in `test.location` every `test.retry_interval` seconds, using authentication and proxy configuration that is specified by the main connection.
3. If the `test.test_on` value is set to "both", then scenario 1 and 2 will both occur. This is useful for users who will use an HTTP connection to move through their firewall - and thereafter frequently refresh the connection to indicate to their firewall that there is still activity for the purpose of keeping the connection alive.



**Note •**

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.logging.clear

**Description** This command clears the weblink log. Setting this value to anything will clear it, including an empty string.

^JUF, ^JUS, ^JUN, ^JUA, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar, do

| Commands | Details  |
|----------|--|
| setvar   | <p>This command clears the weblink log entires.</p> <p><i>Format:</i> ! U1 setvar "weblink.logging.clear" "value"</p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p> |
| do       | <p>This command clears the weblink log entires.</p> <p><i>Format:</i> ! U1 do "weblink.logging.clear" "value"</p> <p><i>Values:</i> Any string value, including an empty string.</p> <p><i>Default:</i> NA</p>     |



**Example** • This example clears the weblink log entries with an empty string value.

```
! U1 setvar "weblink.logging.clear" ""
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.logging.entries

**Description** This command returns the N number of entries in the weblink log, where N has a maximum value that is set by `weblink.logging.max_entries`.

The weblink log is a collection of events related to connecting to a remote Link-OS™ server. The log entries range anywhere from general status to errors that prevented a successful connection. The log contains entries from all connections and are labeled so that it is clear which log entries are for which connection. Each log entry also contains a timestamp for when it was logged by the system. The newest events will appear at the bottom of the list.

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details  |
|---------------------|--|
| <code>getvar</code> | <p>This command returns a lists of entries in the weblink log.</p> <p><i>Format:</i> ! U1 <code>getvar "weblink.logging.entries"</code></p> <p><i>Values:</i> NA</p> <p><i>Default:</i> ""</p> |



**Example** • This example shows the result from `weblink.logging.entries`:

```
[01-04-2013 08:40:45.655] [conn1.1] HTTP/1.1 404 Not Found
[01-04-2013 08:40:45.659] [conn1.1] Received HTTP code 404 from proxy after
CONNECT
[01-04-2013 08:40:45.660] [conn1.1] Closing connection
[01-04-2013 08:40:45.662] [conn1.1] Failed to connect (SP = 0, CU = 0, UW = 0, AC
= 0, PC = 0)
```



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.logging.max\_entries

**Description** This command specifies the maximum number of individual log entries that will be stored in the `weblink.logging.entries` command.



**Important** • Changes to this command are immediate and may result in some log entries being lost. If there are N log entries currently in the log, the user sets the `max_entries` to M, where M is less than N, the oldest (N-M) log entries will be removed.

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar, getvar, do`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command sets the maximum number of log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 getvar "weblink.logging.max_entries" "value"</code></p> <p><i>Values:</i> 0 - 10000</p> <p><b>Note</b> • Setting the value to 0 disables logging.</p> <p><i>Default:</i> "0"</p> |
| <code>getvar</code> | <p>This command returns the setting for the maximum number of log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 getvar "weblink.logging.max_entries"</code></p>   |
| <code>do</code>     | <p>This command sets the maximum number of log entries that will be stored.</p> <p><i>Format:</i> <code>! U1 do "weblink.logging.max_entries" "value"</code></p> <p><i>Values:</i> 0 - 10000</p> <p><b>Note</b> • Setting the value to 0 disables logging.</p> <p><i>Default:</i> "0"</p>     |



**Example 1** • In this example, `weblink.logging.max_entries` is set to 3:

```
[01-04-2013 08:40:45.659] [conn1.1] Received HTTP code 404 from proxy after  
CONNECT
```

```
[01-04-2013 08:40:45.660] [conn1.1] Closing connection
```

```
[01-04-2013 08:40:45.662] [conn1.1] Failed to connect (SP = 0, CU = 0, UW = 0, AC  
= 0, PC = 0)
```

**Example 2** • In this example, `weblink.logging.max_entries` is set to 2:

`weblink.logging.entries` becomes:

```
[01-04-2013 08:40:45.660] [conn1.1] Closing connection
```

```
[01-04-2013 08:40:45.662] [conn1.1] Failed to connect (SP = 0, CU = 0, UW = 0, AC  
= 0, PC = 0)
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## weblink.printer\_reset\_required

**Description** This command retrieves a "yes" or "no" value indicating whether any of the weblink settings have been modified.

^JUF, ^JUS, ^JUN, ^JUA, and `device.restore_defaults` do not have any effect on this setting.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `getvar`

| Commands            | Details   |
|---------------------|---|
| <code>getvar</code> | <p>This command retrieves whether any of the weblink settings are modified.</p> <p><i>Format:</i> ! U1 <code>getvar "weblink.printer_reset_required"</code></p> <p><i>Values:</i> "yes" or "no"</p> <p><i>Default:</i> "no"</p> |



### Note •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

# weblink.restore\_defaults

**Description** This command defaults, and saves, the weblink branch settings. Any value, including an empty string, will default the weblink branch settings.



**Note** • The entire weblink branch of settings will be defaulted and the settings are saved; however, the weblink connections will not use the new settings until the printer is restarted (e.g. the `weblink.printer_reset_required` SGD will be "yes" after a default).

`^JUF`, `^JUS`, `^JUN`, `^JUA`, and `device.restore_defaults` do not have any affect on this setting.

## Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** `setvar`, `do`

| Commands            | Details   |
|---------------------|---|
| <code>setvar</code> | <p>This command defaults the weblink branch settings.</p> <p><i>Format:</i> ! U1 <code>setvar "weblink.restore_defaults" "value"</code></p> <p><i>Values:</i> Any value, including an empty string, will default the branch</p> <p><i>Default:</i> NA</p> |
| <code>do</code>     | <p>This command defaults the weblink branch settings.</p> <p><i>Format:</i> ! U1 <code>do "weblink.restore_defaults" "value"</code></p> <p><i>Values:</i> Any value, including an empty string, will default the branch</p> <p><i>Default:</i> NA</p>     |



**Example** • These all default the branch:

```
! U1 setvar "weblink.restore_defaults" ""
! U1 setvar "weblink.restore_defaults" "foo"
! U1 do "weblink.restore_defaults" ""
! U1 do "weblink.restore_defaults" "foo"
```



**Note** •

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.

## wlan.ip.port\_json\_config

**Description** This command determines the TCP port number to listen on for JSON configuration packets.

### Supported Devices

- iMZ 220™, iMZ 320™
- QLn220™, QLn320™, QLn420™
- ZT210™, ZT220™, ZT230™

**Type** setvar; getvar

| Commands | Details   |
|----------|---|
| setvar   | <p>This command determines the TCP port number on which to listen for JSON configuration packets.</p> <p><i>Format:</i> ! U1 setvar "wlan.ip.port_json_config" "value"</p> <p><i>Values:</i> Any valid port number except ports already in use or the standard network ports.</p> <ul style="list-style-type: none"> <li>• 0 = disable the port</li> <li>• 1-65535 = port number to listen on.</li> </ul> <p><i>Default:</i> "9200"</p> |
| getvar   | <p>This command retrieves the TCP port number which is listening for JSON configuration packets.</p> <p><i>Format:</i> ! U1 getvar "wlan.ip.port_json_config"</p>   |



**Example** • In this example, the `getvar` command causes the printer to get the TCP port number which is listening for JSON configuration packets.

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



### Note

- For printer support, see *SGD Command Support on page 1151*.
- For details on SGD command structure, see *Command Structure on page 636*.







**Zebra Technologies Corporation**

Zebra Technologies Corporation  
475 Half Day Road, Suite 500  
Lincolnshire, IL 60069 USA  
T: +1 847 634 6700  
Toll-free +1 866 230 9494  
F: +1 847 913 8766

**Zebra Technologies Europe Limited**

Dukes Meadow  
Millboard Road  
Bourne End  
Buckinghamshire, SL8 5XF, UK  
T: +44 (0)1628 556000  
F: +44 (0)1628 556001

**Zebra Technologies Asia Pacific, LLC**

120 Robinson Road  
#06-01 Parakou Building  
Singapore 068913  
T: +65 6858 0722  
F: +65 6885 0838

<http://www.zebra.com>