

Application Note: 2457198.190972

Date: 6/21/2015

Topic: Creating your own Weblink endpoint.

Overview: A description of to how to use the Weblink feature in Link-OS printers with your own WebSocket server.

Introduction:

This document describes how to configure a standard WebSocket server to communicate to Zebra Link-OS printers. It contains information showing how to configure a WebSocket server and create initial communications to a printer.

Target Audience:

Developers and IT personnel who wish to:

- Print labels, invoices, receipts, or encode RFID tags from their web pages.
- Create a secure and encrypted connection to Zebra printers from either an on premise a cloud based webserver.

This document is intended for users who have a working knowledge of WebSockets, seeking to create a WebSockets connection utilizing PHP and .NET and other development environments.

This document does not cover Zebra Card printers.

Key Considerations:

This document does not cover all potential issues. Network configuration, firewalls, proxies, and other considerations still have to be accounted for when setting up printers to communicate via WebSockets. See the 'Using Weblink' section of the '[Zebra Programming Guide for SGD, ZPL, and Weblink](#)' document for information on how to get the printer's Weblink log in order to help troubleshoot connectivity issues.

Creating your own Weblink endpoint

Server Configuration

To be able create a websockets connection with the Link-OS printer, the server you choose must support WebSockets. In addition, there are important steps you'll need to take to in order to properly connect from the server to the printer. These steps will allow you to communicate from the server to the printer over the "Raw" or Configuration channels.

The Weblink feature within the printer uses a WebSocket implementation compliant with RFC 6455 (<https://tools.ietf.org/html/rfc6455>). However, the printer only supports Binary frames. If the printer receives text frames it will terminate the connection with response code 1003 indicating that only binary frames are acceptable.

The Weblink implementation on the printer also requires that the HTTP header 'Content-Length' be set to 0 when the upgrade response is sent back to the printer. Typically server implementations will allow the server-side application to provide additional headers or override existing headers. This may be required for your application when the WebSocket upgrade is performed. See the example HTTP handshake in the next section for an example.

SSL Configuration

The printer will only communicate via HTTPS. There are a few configuration settings that are required for the printer to connect. Ensure that your server's SSL configuration at least meets these requirements:

- The server must support TLS v1.x
- The server must support one of the following two ciphers:
 - TLS_RSA_WITH_AES_128_CBC_SHA
 - TLS_RSA_WITH_AES_256_CBC_SHA
- A certificate generated and signed by the Zebra Certificate Authority
 - The process for obtaining a certificate is documented in the attached document.

Double click on the clip image to the right to open the file: 

If there are any issues when connecting, be sure to review the 'Using Weblink' section of the ['Zebra Programming Guide for SGD, ZPL, and Weblink'](#) document. It contains tips that are helpful when troubleshooting SSL connection issues.

Initial WebSocket Handshake

The traffic between the printer and the server should look very similar to the following. If it does not, you will likely not be able to connect the printer. See the 'Using Weblink' section of the ['Zebra Programming Guide for SGD, ZPL, and Weblink'](#) document for information on how to get the printer's Weblink log in order to help troubleshoot connectivity issues. For questions about the HTTP headers seen in the following example, please see RFC 6445 (<https://tools.ietf.org/html/rfc6455>) for more information.

Initial HTTP WebSocket request from the printer

```
GET /yourendpoint HTTP/1.1
Host: your-host:8443
Accept: */*
Cookie: ZEBRA_ID=out-of-ink
Sec-WebSocket-Key: 14Wn1K96GOztjwj5Vj/k1w==
Sec-WebSocket-Protocol: v1.weblink.zebra.com
Sec-WebSocket-Version: 13
Upgrade: websocket
Connection: Upgrade
```

Response from the server to the printer

```
HTTP/1.1 101 Switching Protocols
Content-Length: 0
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: DQ+fjKov3CczM5V22b656k+eA8I=
Sec-WebSocket-Protocol: v1.weblink.zebra.com
```

At this point the printer should have successfully connected.

Once the Main channel is connected

When the main channel (a channel is essentially another name for a connection) is connected the printer will send the server the Zebra Discovery Packet via a JSON message. The Zebra Discovery Packet provides configuration, OS version and other current status information about the printer.

Here is a sample of the Zebra Discovery Packet JSON message (content trimmed from its original length for brevity):

```
{  
  "discovery_b64" : "OiwuBAIBAAFaQlIAAFgAAAA ..."  
}
```

If the printer is configured to send alerts via Weblink (see the section below titled Unsolicited Alerts) alerts will be sent by the printer over the Main channel. They can either be parsed or ignored depending on your needs.

The main channel is also used to open one of two other channels: Raw and Configuration. The Raw channel is similar to the TCP Raw port that is typically listening on port 9100, or any connections via USB, Bluetooth, Serial, or Parallel. The Raw channel will accept any ZPL, CPCL, SGD commands, etc. that are supported by the TCP, USB, Serial, connections.

The Configuration channel is similar to the TCP Raw port that is typically listening on port 9200. The Configuration channel will only accept JSON formatted SGD-like commands that are used to modify or read the printer's settings.

Depending upon your use case, you will need to open one of the two channels (or both at the same time). To do so, the following message will need to be sent to the printer via the Main channel. Substitute in the correct channel for your use case. Note that if you wish to open both channels, one request must be done for each channel request. They cannot be combined into a single request.

How to request the Raw Channel

```
{  
  "open" : "v1.raw.zebra.com"  
}
```

How to request the Configuration Channel

```
{  
  "open" : "v1.config.zebra.com"  
}
```

Each request to open a new channel will result in the printer opening a totally new connection. To determine the type of the channel you may look at the WebSocket protocol string or the initial JSON message sent by the channel. The Raw and Configuration channels do not send the Zebra Discovery Packet, but instead send the details of the new connection/channel. Here is an example message after connecting the Raw channel:

```
{
  "unique_id" : "XXXYYZZZ",
  "channel_name" : "v1.raw.zebra.com",
  "channel_id" : "2"
}
```

The 'channel_id' field in the message is the unique ID given to that channel. It can be useful when troubleshooting printer Weblink issues as the ID shows up in the Weblink log (See [Zebra Programming Guide for SGD, ZPL, and Weblink](#) for more information on how to view the Weblink log).

Reading a Printer Setting via SGD on the Raw Channel

The following example details how to read the printer's friendly name:

1. Point the printer's weblink location to your server and reboot the printer. Note that the server DNS name must match that of the Common Name field within the server SSL certificate.
2. Wait for the Main channel to connect
3. Send the following JSON message to the printer via the Main channel:

```
{
  "open" : "v1.raw.zebra.com"
}
```

4. Wait for the Raw channel to connect
5. Send the following message to the printer via the Raw channel (insert the actual carriage return/line feed characters in place of <CARRIAGE RETURN><LINE FEED>):

```
! U1 getvar "device.friendly_name" <CARRIAGE RETURN><LINE FEED>
```

6. The printer should respond with the friendly name

Reading a Printer Setting via JSON on the Configuration Channel

The following example details how to read the printer's friendly name:

1. Point the printer's weblink location to your server and reboot the printer. Note that the server DNS name must match that of the Common Name field within the server SSL certificate.
2. Wait for the Main channel to connect
3. Send the following JSON message to the printer via the Main channel:

```
{  
  "open" : "v1.config.zebra.com"  
}
```

4. Wait for the Configuration channel to connect
5. Send the following message to the printer via the Configuration channel:

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

6. The printer should respond with the friendly name in a JSON message

Closing Connections

To close the connection from the server, simply use the WebSocket library for your server to cleanly close the connection. This will likely require a close code. Typically the close code used for normal closures is 1000.

Unsolicited Alerts

The printer can send unsolicited alerts over the Main Weblink channel if the printer is configured to do so. You can either register for individual alerts or register for all of them. The following is how to register for alerts and the response to expect when an Unsolicited Alert is send over the Main Channel. Note that the 'configure_alert' payload is that of the ZPL ^SX command and the alerts.add SGD. Please see the [Zebra Programming Guide for SGD, ZPL, and Weblink](#) for more information.

How to Register for a Single Alert

```
{
  "configure_alert" : "HEAD OPEN,SDK,Y,Y,,,N"
}
```

How to Register for All Alerts

```
{
  "configure_alert" : "ALL MESSAGES,SDK,Y,Y,,,N"
}
```

Expected Messages When the Printer Head is Opened

```
{
  "alert" :
  {
    "unique_id" : "XXXYYYZZZ",
    "time_stamp" : "2015-06-09 03:38:12",
    "type_id" : "ERROR",
    "condition_id" : "HEAD OPEN",
    "condition_state" : "SET",
    "type" : "ERROR CONDITION",
    "condition" : "HEAD OPEN"
  }
}
```

Expected Messages When the Printer Head is Closed

```
{
  "alert" :
  {
    "unique_id" : "XXXYYYZZZ",
    "time_stamp" : "2015-06-09 03:38:12",
    "type_id" : "ERROR",
    "condition_id" : "HEAD OPEN",
    "condition_state" : "CLEAR",
    "type" : "ERROR CONDITION",
    "condition" : "HEAD OPEN"
  }
}
```

The 'condition_id' and 'condition_state' indicate the Alert name and if it was "set" or "cleared". They will always be returned in English regardless of the printer's language

configuration. However, 'condition' will be translated similar to the printer front panel alerts. The 'unique_id' indicates the device.unique_id of the printer (sometimes referred to as the printer serial number)

Ping / Pong

The printer will send a PING message roughly every 60 seconds. The server needs to respond with a PONG per the requirements detailed in RFC 6455 (<https://tools.ietf.org/html/rfc6455>) otherwise after three failed PING attempts the printer will disconnect and attempt to reconnect.

Document Control

Version	Date	Description
1.0	6/21/2015	Initial Release

Disclaimer

All links and information provided within this document are correct at time of writing.

Created for Zebra Global ISV Program by Zebra Development Services.