Weblink / WebSocket Endpoint Configuration

Introduction

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

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 webserver or a cloud based webserver.

You must have a working knowledge of WebSockets and want to create a WebSocket connection using PHP, .NET, or other development environments.

Note: 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.

For information on how to get the printer's Weblink log in order to help troubleshoot connectivity issues, see the 'Using Weblink' section of the 'Zebra Programming Guide for SGD, ZPL, and Weblink' document.

Creating Your Own Weblink Endpoint

This section contains the following sections:

- Server Configuration
- SSL Configuration
- Initial WebSocket Handshake

Server Configuration

To create a WebSocket connection with the Link-OS printer, the server you choose must support WebSockets. In addition, there are important steps you'll need to perform in order to properly connect from the server to the printer. These steps 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. However, the printer only supports Binary frames. If the printer receives text frames, it terminates 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 allow the server-side application to provide additional headers or override existing headers. This is may be required for your application when the WebSocket upgrade is performed. See the example HTTP handshake in the next section for more 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 meets the following 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 Appendix describes the process for obtaining a certificate.

If there are any issues when connecting, 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. For information on how to get the printer's Weblink log in order to help troubleshoot connectivity issues, see the 'Using Weblink' section of the 'Zebra Programming Guide for SGD, ZPL, and Weblink' document. For questions about the HTTP headers seen in the following example, please see RFC 6455.

Requesting Your Printer's Configuration

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 sends 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 Unsolicited Alerts section), the printer sends alerts 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 accepts 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 only accepts JSON formatted SGD-like commands that are used to modify or read the printer's settings.

Depending upon your use case, you need to open one of the two channels (or both at the same time). To do so, the following message needs to be sent to the printer via the Main channel. Substitute in the correct channel for your use case.

Note If you wish to open both channels, you must request one for each channel request. You cannot combine 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" : "XXXYYYZZZ",
"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 (For more information on how to view the Weblink log, see the Zebra Programming Guide for SGD, ZPL, and Weblink.

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: 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}
```

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 likely requires a close code. Typically the close code used for normal closures is 1000.

Setting Alerts

This section contains information about alerts.

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 sent over the Main Channel.

Note: The 'configure_alert' payload is that of the ZPL ^SX command and the alerts.add SGD. For more information, see the the Zebra Programming Guide for SGD, ZPL, and Weblink.

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 always return in English, regardless of the printer’s language configuration. However, ‘condition’ is 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 sends a PING message roughly every 60 seconds. The server needs to respond with a PONG, per the requirements detailed in RFC 6455; otherwise, after three failed PING attempts, the printer disconnects and attempts to reconnect.

Appendix

How to Create a Weblink Certificate

1. Download and install the latest version of [Open SSL](#).
2. Create a directory named `zebra_certs`.

   This directory may reside anywhere you choose (desktop, etc.).
3. From the Start menu, choose "run" and type `cmd.exe`.

   This opens a DOS prompt.

   Note: This step requires that you are an administrator.
4. Navigate to your zebra_certs directory. Run the following commands from this directory:
   - Type: `set RANDFILE=.rnd`
   - On the command line, type `openssl`, and then press Enter.
5. Enter the following command and fill in the fields based on the information provided below:
   - Type: `genrsa -out zserver.abccompanyinc.com.key 2048`, and then press Enter.

   Note: `zserver.abccompanyinc.com` = full DNS name of the server. This command generates the key and is part of the security for the server communications. DO NOT give this information out to anyone.
   - Enter the following command and fill in the fields based on the following information:

   ```
   req -new -subj
   "/C=xx/ST=yyyyy/L=aaaaa/O=jjjjj/OU=rrrrrr/emailAddress=ss sss/
   CN=uuuuu" -key uuuuu.key -out uuuuu.csr
   ```

   xx is the two digit Country Code
   aaaaa is the City or town name
   jjjjj is the Organization or company name
   rrrrrr is the Organizational unit name
   sssss is the contact email address for the certificate creator
   uuuuu is the full DNS name of the server

   Note: The DNS name must match the DNS name supplied to the printer as the location URL.

   Example: Listed below is an example of a complete certificate creation request.

   ```
   req -new -subj "/C=US/ST=Illinois/L=Anytown/O=ABC Company
   Inc/OU=IT
   Team/emailAddress=John@abccompanyinc.com/CN=zserver.abcc
   ompanyinc.com" -key zserver.abccompanyinc.com.key -out
   zserver.abccompanyinc.com.csr
   ```
6. Email the certificate file (.csr file) to softpm@zebra.com.

   The certificate will be signed and sent back to you.

7. Copy the zip file containing the signed certificate files to the zebra_certs directory.
8. Extract the signed certificate files into the same directory.
9. Enter the following command and fill in the fields based on the information provided below:

   ```
   pkcs12 -export -in zserver.abccompanyinc.com.cer -inkey
   zserver.abccompanyinc.com.key -out
   zserver.abccompanyinc.com.p12 -name tomcat -CAfile
   ZebraCAChain.cer -caname root –chain
   ```

   Where `zserver.abccompanyinc.com` is the full DNS name of the server

   Note: This step converts the certificate and asks you to set a passkey.
10. Enter a standard alphanumeric passkey, but do not include any special characters (for example, do not use characters such as $, %, &, or @).

    Note: The passkey should be something easy to remember, but should not be distributed to anyone.
11. Configure your server to use the passkey (created in step 9) and the certificate file. If you are using a Tomcat server, edit the Tomcat server.xml in the

    ```
    %TOMCAT_INSTALL_LOCATION%\conf directory to use the new
    key/cert by modifiying the ssl connector as follows.
    ```

    - Edit the XML document to include the following text within the <Service> XML block.

    ```
    <Service name="Catalina">
    …
    <Connector SSLEnabled="true" acceptorThreadCount="5"
    clientAuth="want" keyAlias="tomcat"
    keystoreFile="conf/zserver.abccompanyinc.com.p12"
    keystorePass="YourPasskey" keystoreType="pkcs12"
    maxConnections="-1" maxThreads="2500" port="443"
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    scheme="https" secure="true" sessionTimeout="0"
    socket.soKeepAlive="true" sslProtocol="TLS"/>
    …
    </Service>
    ```

    - Where `zserver.abccompanyinc.com` is the full DNS name of the server.
    - Where YourPasskey = passkey from Step 9.
12. Run the following command from the zebra_certs directory:

    ```
    %> keytool -importcert -file ZebraCAChain.cer –keystore
    "%JRE_HOME%\lib\security\cacerts" -alias "ZebraCAChain"
    ```

    Note: The default password for the Java cacert keystore is `changeit`.

    Note: Run this command for the same JRE in use by the Tomcat instance being used.

Document Control

Version	Date	Description
1.0	6/29/2015	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.