

FX CONNECT



ZEBRA

User Guide

FX CONNECT USER GUIDE

MN-003950-01EN

Revision A

April 2020

Copyright

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2019-2020 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

COPYRIGHTS & TRADEMARKS: For complete copyright and trademark information, go to www.zebra.com/copyright.

WARRANTY: For complete warranty information, go to www.zebra.com/warranty.

END USER LICENSE AGREEMENT: For complete EULA information, go to www.zebra.com/eula.

For Australia Only

For Australia Only. This warranty is given by Zebra Technologies Asia Pacific Pte. Ltd., 71 Robinson Road, #05-02/03, Singapore 068895, Singapore. Our goods come with guarantees that cannot be excluded under the Australia Consumer Law. You are entitled to a replacement or refund for a major failure and compensation for any other reasonably foreseeable loss or damage. You are also entitled to have the goods repaired or replaced if the goods fail to be of acceptable quality and the failure does not amount to a major failure.

Zebra Technologies Corporation Australia's limited warranty above is in addition to any rights and remedies you may have under the Australian Consumer Law. If you have any queries, please call Zebra Technologies Corporation at +65 6858 0722. You may also visit our website: www.zebra.com for the most updated warranty terms.

Terms of Use

- Proprietary Statement

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

- Product Improvements

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

- Liability Disclaimer

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

- Limitation of Liability

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

Revision History

Changes to the original manual are listed below:

Change	Date	Description
-01 Rev A	4/2020	Initial release

Table of Contents

Copyright	3
For Australia Only	3
Terms of Use	3
Revision History	4

FX Connect

Introduction	6
Using FX Connect	6
Running Inventory on FX Connect	14
Example JSON format of Tag Data	14
Example Key-Value Pair Format of Tag Data	14
Configuring the HTTP Post Server	16
Configuring the HTTP Proxy Server	17
Configuring USB HID	17
Configuring the TCP/IP Socket	18
Configuring the USB Flash Drive	21

Using FX Connect

Introduction

FX Connect is a licensed feature which enables users to easily collect data from FX series RFID readers, namely the FX7500 and FX9600. Data is pushed to the host PC in keystrokes via USB-HID or HTTP POST in a hassle free manner. No knowledge of APIs or application development is required to receive RFID data from the reader.

Refer to the FX Series RFID Fixed Reader Licensing Management User Guide for detailed licensing information.

Starting FX Connect

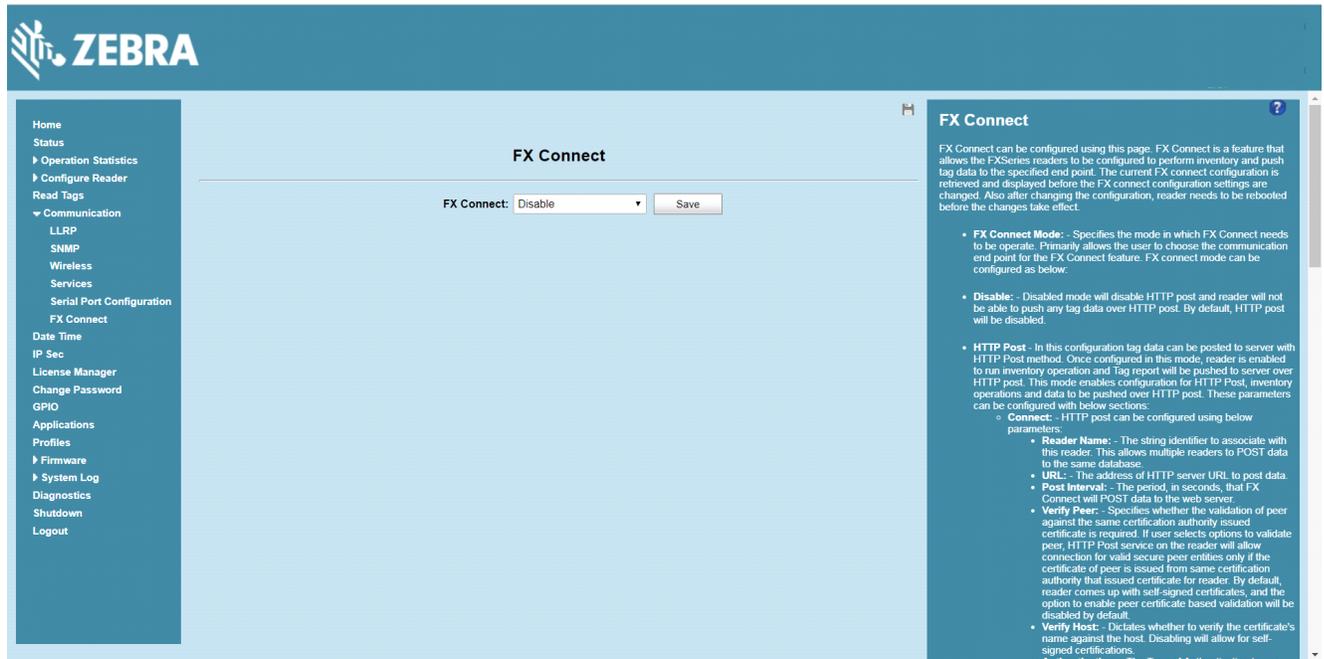
1. Open a web browser to connect to the FX reader using the host name or IP address. (See [Quick Start on page 15](#) for startup instructions.)
2. Click **Communication > FX Connect**.



NOTE If no valid license is installed yet, the screen displays message to remind users to obtain a valid license. Refer to the FX Series RFID Fixed Reader Licensing Management User Guide for information about how to enable FX Connect via a license.

If a valid license is installed, it is displayed in the FX Connect console.

Figure 1 FX Connect



3. Select the FX Connect drop-down arrow to select an option.
 - a. **Disable** - By default, FX Connect is disabled.
 - b. **HTTP Post** - This option enables the reader to push Inventory Tag data in JSON format to the web server using the HTTP Post method. The web server could be running on local network or in the cloud. See [Table 1](#) for field information.
 - c. **Keyboard Emulation** - This option allows the user to push data over USB HID. It enables the reader to send RFID data to an application running on a PC host connected to the reader via the USB client port. Any application that can receive keyboard input is able to receive RFID data from the reader in this mode because the reader uses Keyboard HID emulation to send data to the host PC. If this option is selected and the user starts the Inventory, tag data is shown in a key-value pair in the selected text editor (Notepad, MS Excel, etc.). See [Table 2](#) for field information.
 - d. **TCP/IP Socket** - As in HTTP POST, this option also enables the reader to push Inventory Tag data in JSON format to the defined TCP/IP socket port. Any client socket application connected to the reader with defined port can receive Tag data. See [Table 3](#) for field information.
 - e. **USB Flash Drive** - When this option is selected, the tag data is written to a specific file in attached USB Flash drive in Key Value Pair format. This mode does not have any specific configurable parameters. See [Table 4](#) for field information.
4. Select **Save** to save the configuration settings for FX Connect.

Figure 2 HTTP Post Screen

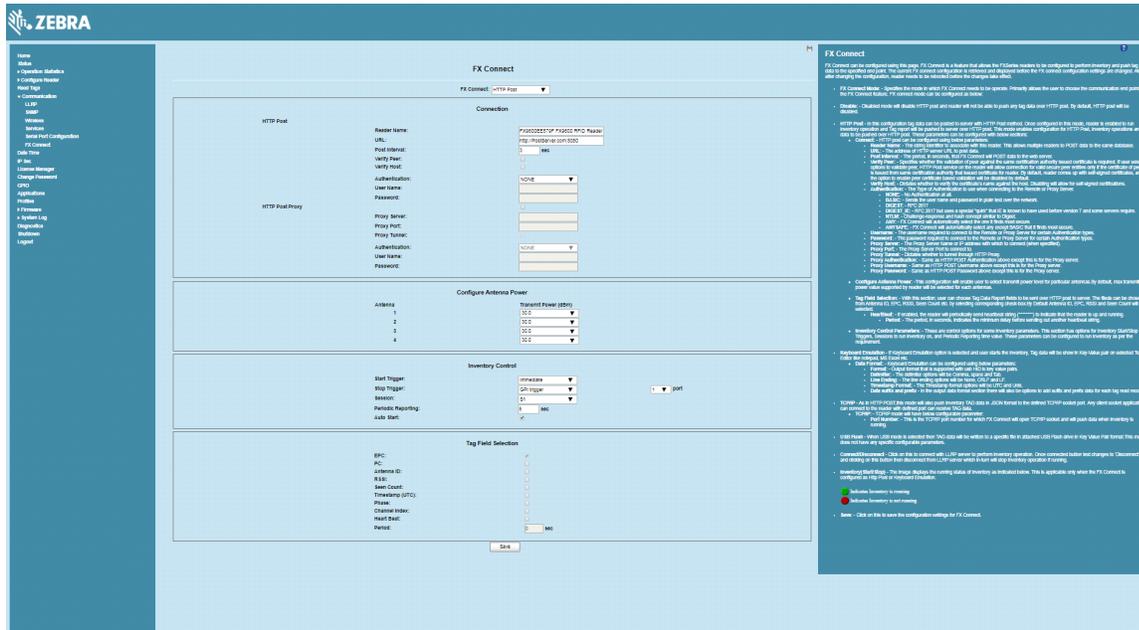


Table 1 HTTP Post Configurable Options

Option	Description
HTTP Post - This section displays the configurable parameters for the HTTP Post server.	
Reader Name	Reader name is the text string that appears in Post data to identify the reader. See Configuring the HTTP Post Server on page 16 for details on how to set up a web server to test this feature.
URL	HTTP Post Server URL to receive post data.
Post Interval	The period, in seconds, that FX Connect posts data to the web server.
Verify Peer	Specifies whether or not peer validation is required. If the user selects the option to validate peer, the HTTP Post service on the reader allows connection for valid secure peer entities only if the certificate of peer is issued from the same certification authority that issued the certificate for the reader. By default, the reader issues self-signed certificates and the option to enable peer certificate-based validation is disabled.
Verify Host	Dictates whether to verify the certificate's name against the host. Disabling this option allows self-signed certifications.

Table 1 HTTP Post Configurable Options (Continued)

Option	Description
Authentication	<p>The type of authentication to use when connecting to the remote or proxy server.</p> <ul style="list-style-type: none"> • NONE: No authentication at all. • BASIC: Sends the user name and password in plain text over the network. • DIGEST: RFC 2617. • DIGEST_IE: RFC 2617 but uses a special quirk that IE is known to have used before version 7 and some servers require. • NTLM: Challenge-response and hash concept similar to Digest. • ANY: FX Connect will automatically select the one it finds most secure. • ANYSAFE: FX Connect will automatically select any except BASIC that it finds most secure.
User Name	The user name required to connect to the Remote or Proxy Server for certain Authentication types.
Password	The password required to connect to the Remote or Proxy Server for certain Authentication types.
<p>HTTP Post Proxy: If reader is behind a proxy server then select the check box. This provides proxy server related parameters. For more details on how to configure a proxy server see Configuring the HTTP Post Server on page 16.</p>	
Proxy Server	The Proxy Server Name or IP address with which to connect (when specified).
Proxy Port	The Proxy Server port to connect to.
Proxy Tunnel	Dictates whether to tunnel through HTTP Proxy.
Authentication	Same as HTTP Post Authentication but for the proxy server.
User Name	Same as HTTP Post User Name but for the proxy server.
Password	Same as HTTP Post Password but for the Proxy server.
Configure Antenna Power	Enables the user to select the transmit power level for particular antennas. By default, the maximum transmit power value supported by the reader is selected for each antenna.

Table 1 HTTP Post Configurable Options (Continued)

Option	Description
Inventory Control Parameters	<p>These are control options for some inventory parameters. This section has options for inventory start/stop triggers, sessions on which to run inventory, and periodic reporting time value. These parameters can be configured to run inventory as per the requirement. If the user selects the auto start option, inventory starts automatically upon boot up (based on the start trigger chosen).</p>
Tag Field Selection	<p>In this section the user can choose Tag Data Report fields to be sent over HTTP Post to the server. The available fields to chosen are: EPC, PC, Antenna ID, RSSI, Seen Count, etc. by selecting the corresponding check-box. By default, EPC, PC, Antenna ID, and Seen Count are selected.</p> <p>The Tag Field Selection descriptions are as follow:</p> <ul style="list-style-type: none"> • EPC - The Tag EPC Data field. • PC - PC BITS. • Antenna ID - Antenna ID on which Tag is read. • RSSI - The Peak RSSI value. • Seen Count - The number of times the Tag is read (in-case of periodic reporting). • Timestamp - Timestamp when Tag is seen. • Phase - Phase information reported by the reader when the tag is seen. • Channel Index - Index of the first channel when the tag is seen. • Heart Beat - Heartbeat string to indicate that the reader is UP and running. <p>Note: Heart Beat, when enabled, causes the reader to periodically send a heartbeat string (******) to indicate that the reader is up and running.</p> <p>The Period for heartbeat, in seconds, indicates the minimum delay before sending out another heartbeat string.</p>

Figure 3 Keyboard Emulation Screen

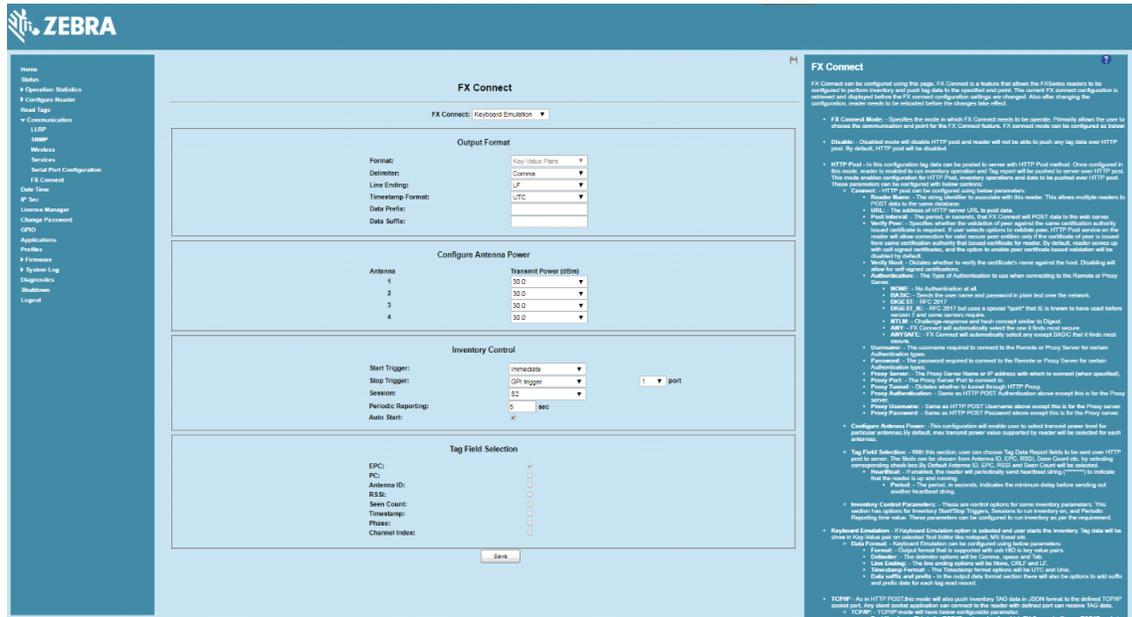


Table 2 Keyboard Emulation Configurable Options

Option	Description
Output Format - This section displays the configurable parameters for the Keyboard Emulation.	
Format	Output format that is supported with USB HID is key-value pairs.
Delimiter	The delimiter options are comma, space, and tab.
Line Ending	The line ending options are None, CRLF, and LF.
Timestamp Format	The timestamp format options are UTC and Unix.
Data Prefix/Data Suffix	The user can add a prefix and suffix for each tag read record. Note: Data Prefix and Data Suffix should be in a character sequence only.
Configure Antenna Power	See Table 1 .
Inventory Control Parameters	See Table 1 .
Tag Field Selection	See Table 1 (defaults vary).

Figure 4 TCP/IP Socket Screen

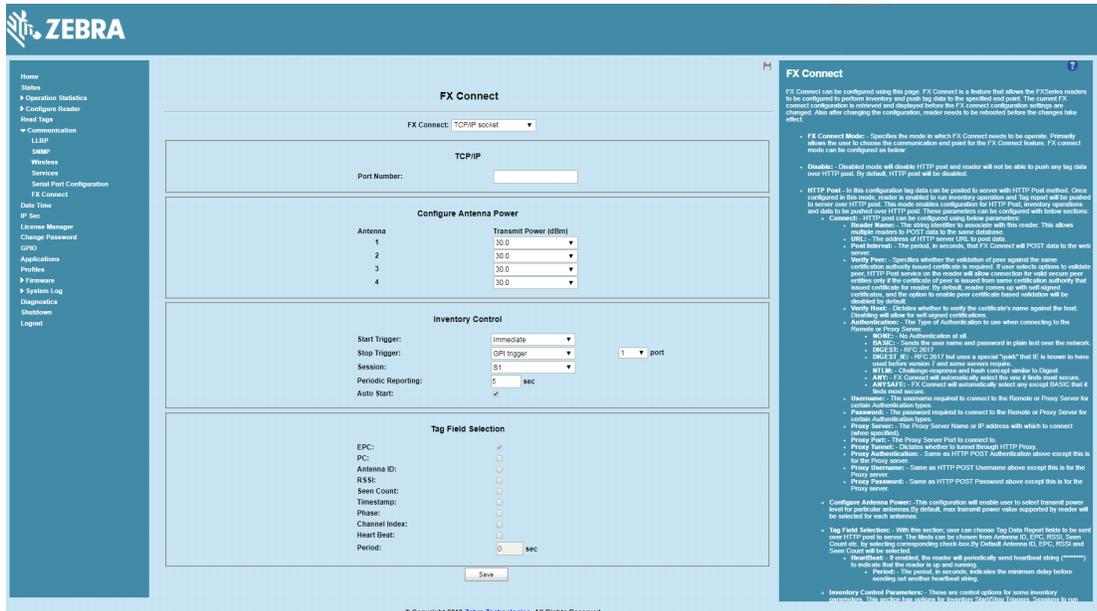


Table 3 TCP/IP Socket Configurable Options

Option	Description
TCP/IP - This section displays the configurable parameters for the TCP/IP.	
Port Number	This is the TCP/IP port number for which FX Connect will open TCP/IP socket and push data when inventory is running.
Configure Antenna Power	See Table 1 .
Inventory Control Parameters	See Table 1 .
Tag Field Selection	See Table 1 (defaults vary).

Figure 5 USB Flash Drive

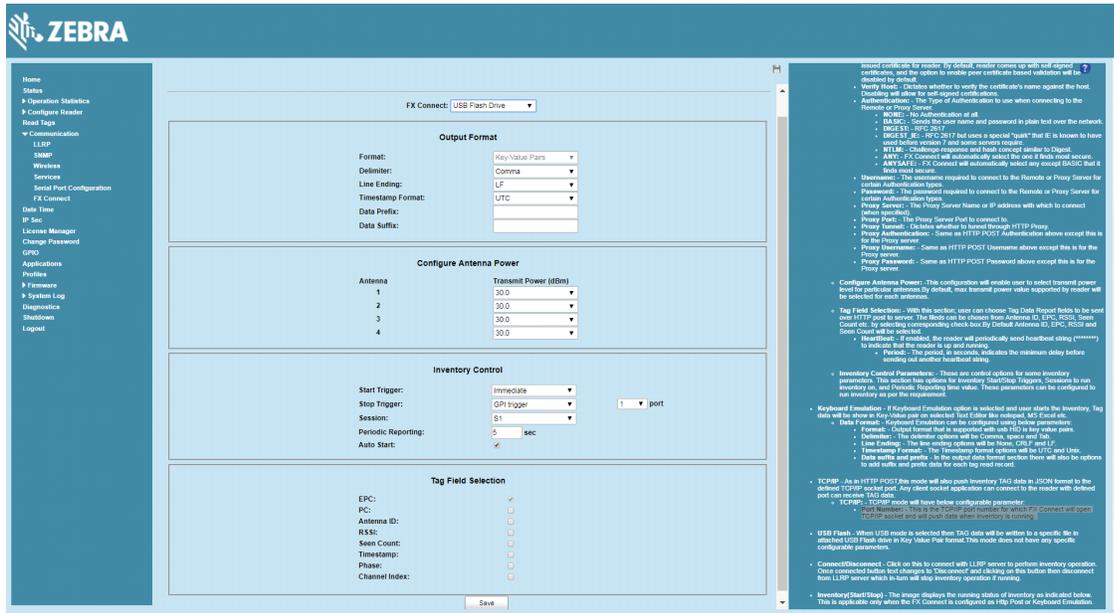


Table 4 USB Flash Drive Configurable Options

Option	Description
Output Format	This section displays the configurable parameters for the USB Flash Drive.
Format	Output format that is supported with USB HID is key-value pairs.
Delimiter	The delimiter options are comma, space, and tab.
Line Ending	The line ending options are None, CRLF, and LF.
Timestamp Format	The timestamp format options are UTC and Unix.
Data Prefix/Data Suffix	The user can add a prefix and suffix for each tag read record. Note: Data Prefix and Data Suffix should be in a character sequence only.
Configure Antenna Power	See Table 1.
Inventory Control Parameters	See Table 1.
Tag Field Selection	See Table 1 (defaults vary).

Running Inventory on FX Connect

To start the inventory operation for the selected and configured Output mode:

1. From the FX Connect console select **Connect**.



NOTE **Connect** appears after the user selects **Save**.

Connect changes to **Disconnect** after a successful connection.

- a. The inventory operation begins per the configured Start Trigger setting.
 - b. The inventory status light turns green when inventory beings running. It turns red when the inventory stops per the configured Stop Trigger setting.
2. Select **Disconnect**.



NOTE **Disconnect** changes to **Connect** after a successful disconnection.

Example JSON format of Tag Data

For the HTTP POST and TCP/IP Socket mode, Tag Data is reported in JSON format. Below is the example of JSON format (with selecting all Tag Field):

```
{ "reader_name": "FX9600FB37EE FX9600 RFID Reader", "mac_address": "84:24:8D:EF:B2:F6",
  "tag_reads": [{"epc": "8DF00000000000007CCC62", "pc": "3000", "antennaPort": "1", "peakRssi": "-58", "seenCount": "4", "timeStamp": "30/3/2020
17:7:23:67", "phase": "0.00", "channelIndex": "1", "isHeartBeat": "false"}, {"epc": "30304035A880C800001235CD",
  "pc": "3000", "antennaPort": "1", "peakRssi": "-48", "seenCount": "2", "timeStamp": "30/3/2020
17:7:23:847", "phase": "0.00", "channelIndex": "1", "isHeartBeat": "false"}]}
```

Below is the example of Heartbeat JSON data, which is reported at a defined time interval as per the configuration.

```
{ "reader_name": "FX9600FB37EE FX9600 RFID Reader", "mac_address": "84:24:8D:EF:B2:F6",
  "tag_reads": [{"epc": "*****", "pc": "0", "antennaPort": "0", "peakRssi": "0", "seenCount": "0", "timeStamp": "3
0/2/2020 17:7:49:416565", "phase": "0", "channelIndex": "0", "isHeartBeat": "true"}]}
```

Example Key-Value Pair Format of Tag Data

For the Keyboard Emulation and USB Flash Drive mode in FX Connect, Tag Data is reported in Key-Value Pair format. Key-Value Pair format mostly depends on the delimiter and line-ending configuration. The Key-Value Pair format varies as per the choice of these configuration option.

Apart from these, there is one special case; if EPC only is selected in **Tag Field Selection**, then Key-Value Pair will have data output without any meta-data as shown in example 1 & 3 below.

1. With **EPC** only in Tag Field Selection and **Line Ending** as None.

```
ad72120544894bb660000059,e280689000000000182a55b1,303800000333f10000000039,303800000333f10000000039,ad72120544894bb660000059,e280689000000000182a55b1,
```

2. With more selections in Tag Field Selection and **Line Ending** as None.

```
epc=303800000333f10000000039,pc=3000,antennaport=2,peakrssi=-60,seencount=1,timestamp=15112019610110,phase=0.00,channelindex=6,epc=ad72120544894bb660000059,pc=3000,antennaport=2,peakrssi=-59,seencount=1,timestamp=15112019610118,phase=0.00,channelindex=6,epc=e280689000000000182a55b1,pc=3000,antennaport=2,peakrssi=-60,seencount=1,timestamp=15112019610122,phase=0.00,channelindex=6,epc=303800000333f10000000039,pc=3000,antennaport=2,peakrssi=-58,seencount=13,timestamp=1511201961168,phase=0.00,channelindex=8,epc=ad72120544894bb660000059,pc=3000,antennaport=2,peakrssi=-54,seencount=14,timestamp=1511201961169,phase=0.00,channelindex=8,epc=e280689000000000182a55b1,pc=3000,antennaport=2,peakrssi=-59,seencount=14,timestamp=1511201961170,phase=0.00,channelindex=8,
```

3. With **EPC** only in Tag Field Selection and **Line Ending** as LF (Line Feed).

```
303800000333f10000000039,ad72120544894bb660000059,ad72120643c065b4867e00ab,e280689000000000182a55b1,
```

4. With more selections in Tag Field Selection and **Line Ending** as LF (Line Feed).

```
epc=ad72120544894bb660000059,pc=3000,antennaport=2,peakrssi=-60,seencount=1,timestamp=151120196011418,phase=0.00,channelindex=6,epc=303800000333f10000000039,pc=3000,antennaport=2,peakrssi=-58,seencount=1,timestamp=151120196011422,phase=0.00,channelindex=6,epc=e280689000000000182a55b1,pc=3000,antennaport=2,peakrssi=-60,seencount=1,timestamp=151120196011433,phase=0.00,channelindex=6,epc=ad72120544894bb660000059,pc=3000,antennaport=2,peakrssi=-55,seencount=44,timestamp=151120196016418,phase=0.00,channelindex=18,
```



NOTE In all the examples shown in from No. 1 to 4, **Comma (,)** is selected as the Delimiter.

Configuring the HTTP Post Server



NOTE You must have a valid license to run FX Connect. Refer to the FX Series RFID Fixed Reader Licensing Management User Guide for more information.

To run the HTTP Server to receive tag data from FX Connect:

1. Open a web browser to connect to the FX reader using the host name or IP address. (See [Quick Start](#) for startup instructions.)
2. Click **Communication > FX Connect**.
3. Install Python version 2.7 or above.
4. Go to: github.com/BurntSushi/nfldb/wiki/Python-&-pip-Windows-installation and follow the instructions to install Python pip.
5. Open command shell.
6. Install Flask by typing the command: `pip install Flask`.
7. Save the text below as `postServer.py`.

```
from flask import Flask, request
app = Flask(__name__)

@app.route('/', methods = ['POST', 'GET'])

def message():
    if request.method == 'POST':
        app.logger.info('Request received.')
        app.logger.info('Url: %s', request.url)
        app.logger.info('Data: %s', (request.data).decode('utf-8'))
        app.logger.info('Is JSON: %s', request.is_json)
    else:
        app.logger.info('GET request received.')
    return 'OK\n'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port='5001', debug=1)
```



NOTE The script above runs HTTP server on port 5001; if needed, change the port number.

8. To see the HTTP Post output, run the command: `python postServer.py`. This starts the HTTP server which receives tag data from FX Connect and displays the data in the command shell.

Configuring the HTTP Proxy Server

To setup the Squid proxy server on an Ubuntu machine:

1. Install, start, and enable Squid on the Ubuntu machine using the following commands.


```
$ sudo apt -y install squid
$ sudo systemctl start squid
$ sudo systemctl enable squid
```
2. The squid.conf file is created in: /etc/squid/squid.conf.
3. The default configuration file contains some configuration directives that affect the behavior of the Squid and they need to be configured.

Open the squid.conf file and update the lines below, the **Save**.

- a. Under **rule allowing access from your local networks** add the following statements:

```
acl all src 0.0.0.0/0.0.0.0
acl Safe_ports port 8081          # custom http
```

- b. Under **access permission configuration** add the following statement:

```
http_access allow all
```

- c. Under **access permission configuration** comment the statements:

```
#http_access allow localhost manager
#http_access deny manager
```

4. Restart the Squid with the following command:


```
sudo systemctl restart squid
```
5. The proxy server is now ready.
6. Check the logs in the proxy server with the following command:


```
sudo tail -f /var/log/squid/access.log
```

Configuring USB HID



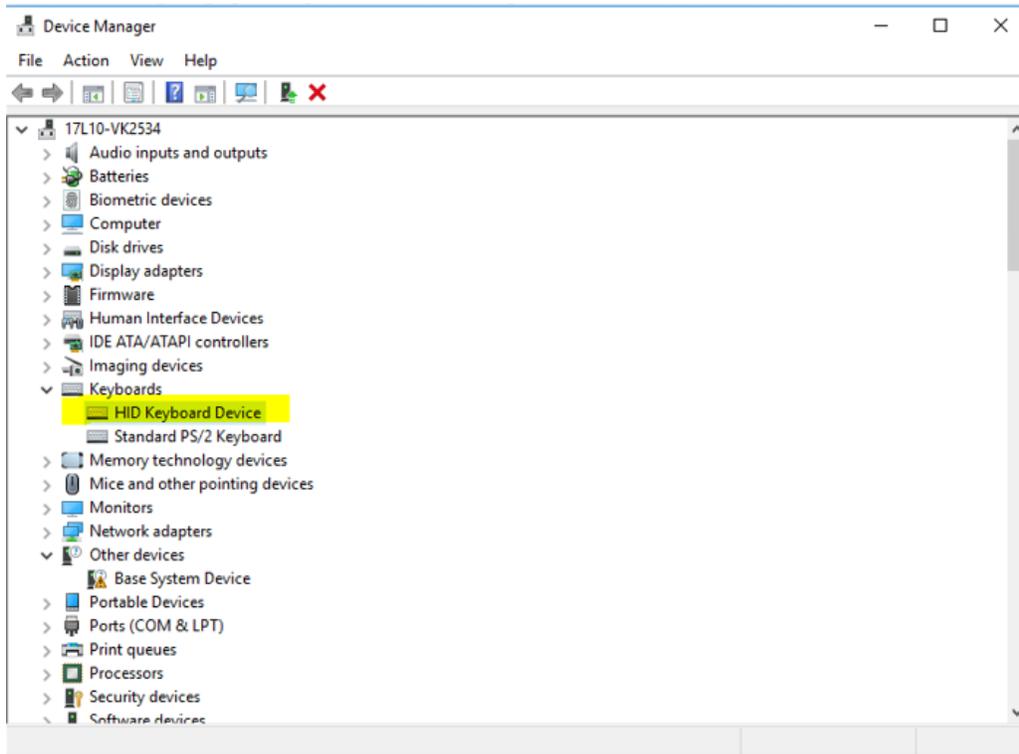
NOTE It is recommended to use two host PCs - one to control the reader through the web console and another to receive tag data.



NOTE Setting the reader to use USB HID disables RNDIS on the USB client port. The reader cannot be accessed using the RNDIS IP Address (169.254.10.1) in this scenario.

To run FX Connect in Keyboard Emulation:

1. Open the reader web console to FX Connect (see [Figure 1 on page 7](#)).
2. Connect the reader to the host machine through the USB Client port (same as RNDIS). See [Figure 7 on page 22](#) and [Figure 10 on page 25](#) for FX7500 and FX9600 USB Client ports. Windows automatically detects the reader as an HID device when inventory starts on FX Connect and enables the driver.

Figure 6 HID Device Detection

3. Open any text editor application or MS Excel to receive push data from the reader. Use your cursor to select the spot where you want to print the tag data.
4. Start the inventory by selecting **Connect** on reader web console.

Tag data is printed in the application at the selected spot.

Configuring the TCP/IP Socket

To use TCP/IP socket for receiving Tag data

1. Run TCP/IP socket client application on host machine. TCP/IP client will read the Tag data sent by the reader. Below is the simple TCP/IP client application which need to be compiled on host machine:

```

/*
 * tcpclient.c - A simple TCP client
 * usage: tcpclient <server IP> <port>
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

```

(continued on next page)

```

#include <netdb.h>
#include <sys/ioctl.h>

#define BUFSIZE 10240

/*
 * error - wrapper for perror
 */
void error(char *msg) {
    perror(msg);
    exit(0);
}

int main(int argc, char **argv) {
    int sockfd, portno, n;
    struct sockaddr_in serveraddr;
    struct hostent *server;
    char *hostname;
    char buf[BUFSIZE];
    struct timeval t;
    int iMode = 1;
    int count = 0;

    /* check command line arguments */
    if (argc != 3) {
        fprintf(stderr, "usage: %s <server IP> <port>\n", argv[0]);
        exit(0);
    }
    hostname = argv[1];
    portno = atoi(argv[2]);

    /* socket: create the socket */
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");

```

(continued on next page)

```

/* gethostbyname: get the server's DNS entry */
server = gethostbyname(hostname);
if (server == NULL) {
    fprintf(stderr, "ERROR, no such host as %s\n", hostname);
    exit(0);
}
printf("Connecting to server : %s\n", server->h_name);

/* build the server's Internet address */
bzero((char *) &serveraddr, sizeof(serveraddr));
serveraddr.sin_family = AF_INET;
bcopy((char *)server->h_addr,
      (char *)&serveraddr.sin_addr.s_addr, server->h_length);
serveraddr.sin_port = htons(portno);

/* connect: create a connection with the server */
if (connect(sockfd, (struct sockaddr *)&serveraddr, sizeof(serveraddr)) < 0){
    printf("Connecting to socket failed.\n");
    close(sockfd);
    error("ERROR connecting");
    return -1;
}
while(1){
    /* print the server's reply */
    bzero(buf, BUFSIZE);
    n = read(sockfd, buf, BUFSIZE);
    if (n < 0){
        error("ERROR reading from socket");
    } else if (n == 0){
        printf("Server Socket closed \n");
        break;
    } else {
        printf("\nMessage: %s", buf);
    }
    //if(count++ > 10)
    //    break;
}
close(sockfd);
return 0;
}

```

2. Once compiled, run this application with reader IP and PORT which is already configured on the reader.
3. After running the application, click on connect button on the reader web console to run inventory under FX Connect.
4. Application will print the received tag data on screen.

Configuring the USB Flash Drive

To get the tag data in USB drive there no configuration is required. The user just need to attach the USB flash drive in reader and click on **Connect** button at FX Connect web console.

The tag data will be pushed to USB flash drive in a file named as current timestamp.

