TRUDU, Laurent

ZebrA Enterprise SERVICES

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Comments |
| 28th January 2019  14th February 2019 | Laurent Trudu  Laurent Trudu | 1.00  1.10 | Initial version  Updated with new Quit mode |
| 20th March 2019 | Laurent Trudu | 2.0 | Added REST WebServices |
|  |  |  |  |

[1. Overview 5](#_Toc4400757)

[1. Architecture 6](#_Toc4400758)

[2. Pre-requisites 7](#_Toc4400759)

[3. PrintConnect Installation 7](#_Toc4400760)

[4. Zebra Enterprise Services Installation 7](#_Toc4400761)

[5. Licence 7](#_Toc4400762)

[2. REST Web Service 8](#_Toc4400763)

[1. How To 8](#_Toc4400764)

[2. URL Format 9](#_Toc4400765)

[3. Return value 11](#_Toc4400766)

[4. Data Encoding 12](#_Toc4400767)

[5. Commands 14](#_Toc4400768)

[1. Connect 14](#_Toc4400769)

[2. ZPL Printing 15](#_Toc4400770)

[3. ZPL Printing with variable data 16](#_Toc4400771)

[4. FileName Print 18](#_Toc4400772)

[5. Line Print 19](#_Toc4400773)

[6. Passthrough Print 20](#_Toc4400774)

[7. Unselect Printer 21](#_Toc4400775)

[3. Browser Intent URL Service 22](#_Toc4400776)

[1. How To 22](#_Toc4400777)

[2. URL Format 22](#_Toc4400778)

[3. Feedback control 22](#_Toc4400779)

[4. Data Encoding 24](#_Toc4400780)

[5. Quit Parameter 26](#_Toc4400781)

[6. Commands 27](#_Toc4400782)

[8. ZPL Printing 27](#_Toc4400783)

[9. ZPL Printing with variable data 28](#_Toc4400784)

[10. FileName Print 29](#_Toc4400785)

[11. Line Print 30](#_Toc4400786)

[12. Passthrough Print 31](#_Toc4400787)

[13. Unselect Printer 32](#_Toc4400788)

[4. Appendix 33](#_Toc4400789)

[1. Licence 33](#_Toc4400790)

[5. Source code 38](#_Toc4400791)

# Overview

The Zebra Enterprise Services expose industrial functionalities to any Android Web Browser.

The first implementation of the Services implements the PrintConnect API, this allows to connect, disconnect and send print jobs to any Zebra LinkOS Printer.

The printing API has been implemented using two different entry points:

* A local REST Web Service  
  It allows any Android browser to send REST commands using standard Web methods, and retrieve the result in a JSON format response.
* An implementation of the Chrome Intent URL Schemes.  
  The Chrome Intent implementation allows any Android browser to send print commands using the intent:// URL Scheme.  
  The results are displayed using Android native Toasts windows (small popup window that appears on top of any application for a user defined duration, then disappears)  
  To enable Chrome Intent protocol, the Zebra Enterprise Services (ZES) will register a new URL commands to the intent:// scheme.  
  Each command will allow the user to do specific actions.

Both implementations can be addressed using standard Web methods.

They rely on Zebra’s PrintConnect Driver to ensure the communication with the Printer.

Thanks to PrintConnect and ZES, any Android browser can communicate with any LinkOS Zebra’s Printer through Bluetooth, Wifi, Ethernet.

More information about PrintConnect can be found on the Zebra’s Website:

<https://www.zebra.com/us/en/products/software/barcode-printers/link-os/print-connect.html>

All the commands of Browser Intent URL Service derive from the PrintConnect Intent API.

For more information, refer to the PrintConnect documentation (in the Intents section you'll find a description of the methods implemented by the BIUS/RESTWs):

<https://www.zebra.com/content/dam/zebra_new_ia/en-us/solutions-verticals/product/Software/Printer%20Software/Link-OS/print-connect/PC-UserGuide-P1082444-001.pdf>

## Architecture

This solution is based on two components: PrintConnect that acts like a printer driver on Android, and ZES that wraps all PrintConnect commands in a Web REST Service or as Chrome Intent URLs.

Chrome, Firefox,Opera,…

Enterprise Services

PrintConnect

Android

Zebra’s LinkOS Printer

## Pre-requisites

Hardware:

* An Android device
* A Zebra’s LinkOS Printer

Software:

* Zebra’s PrintConnect
* ZebraEnterpriseServices.apk

## PrintConnect Installation

To work properly, the PrintConnect driver MUST be installed before using the Browser Intent URL Service.

PrintConnect can be downloaded from the Google Playstore:

<https://play.google.com/store/apps/details?id=com.zebra.printconnect>

Or from the Zebra Website:

<https://www.zebra.com/us/en/products/software/barcode-printers/link-os/print-connect.html>

## Zebra Enterprise Services Installation

Install the Zebra PrintConnect apk on the device.

Install the Zebra Enterprise Services apk on the device.

1. Chrome Intent URLs installation steps:
   * Reboot the device.
   * Launch your enterprise web application in your selected browser
2. REST Web Services manual installation steps:
   * Launch ZES application
   * Start the Service

## Licence

The Licence associated with ZES can be read from the application graphical user interface thanks to the “Licence” button or is available at the end of the current document in the Appendix section ([Licence](#_Licence))

**Please read carefully the Zebra Open Source Software License associated with this project before use**

# REST Web Service

## How To

The printer must be paired using PrintConnect to allow the Web Service to work properly.

To do this, simply install PrintConnect, and then do an NFC pairing with a mobile bluetooth printer or any other kind of pairing (Wifi, Ethernet)

The process is explained in the PrintConnect user guide.

The print commands can be executed using standard REST WS calls on the device local IP address.

* Example of printing ZPL using the window.open API:  
  window.open("http://localhost:8080/?command=printzpl&template=^XA^FO10,10^ADN,300,70^FDTEST^FS^XZ");
* Example of printing a text in single line mode using fetch with base64 encoded data:  
  fetch("http://127.0.0.1:8080/?command=printsingleline&encoding=base64encoded&standardCharsets=UTF\_16&text=//5UAGgAaQBzACAAaQBzACAAbQB5ACAAdABlAHgAdAANAAoAVABvACAAcAByAGkAbgB0AA0ACgBFAG4AYwBvAGQAZQBkACAAaQBuACAAYgBhAHMAZQA2ADQADQAKAFcAaQB0AGgAIABjAGEAcgByAGkAYQBnAGUAIAByAGUAdAB1AHIAbgBzAA0ACgANAAoA").then(response => { if(response.ok) return response.json();}).then(json => document.getElementById("RestWSResponse").innerHTML = "<p>Status : " + json.result + "</p>\n<p>Message : " + json.message + "</p>\n");
* Example of unselecting the printer using XMLHttpRequest:

var xhr = new XMLHttpRequest();

var encodedCommand = encodeURIComponent("unselect");

xhr.open("GET", "http://"+ destIP + ":8080/?command="+ encodedCommand, true);

xhr.onreadystatechange = function (oEvent)

{

if (xhr.readyState === XMLHttpRequest.DONE)

{

if (xhr.status === 200)

{

console.log(xhr.responseText);

document.getElementById("RestWSResponse").innerHTML = "<p>Success : " + xhr.responseText + "</p>\n";

}

else

{

console.log("Error", xhr.statusText);

document.getElementById("RestWSResponse").innerHTML = "<p>Error : " + xhr.statusText + "</p>\n";

}

}

};

xhr.send(null);

For better readability, the REST functions will be showcased using the target URL.

The choice of the method used to execute this URL is left to the user discretion.

## URL Format

The commands are passed in a classic REST WS Call format.

The URL are formed this way:

http://localhost:8080/?command=mycommand&myRESTParameter1=something

The text in dark blue represent the mandatory part of the URL.

The text in green represent the variable members of the URL.

**As we are dealing with URL all the texts or base64Data that you will use MUST be URIEncoded before usage.**

The standard javascript method encodeURIComponent will replace special characters like + or / or = with their % equivalent.

For example, / will be replaced by %2F.

Using URIEncoded values ensure that the content of the data that is passed to the service won’t interfere with the URI format.

Ex:

The following base64 encoded template can’t be used without being encoded:



**The //, + and = characters are reserved characters.**

Here is the result after URI encoding:

%2F%2F5DAFQAfgB%2BAEMARAAsAH4AQwBDAF4AfgBDAFQAfgBeAFgAQQB%2BAFQAQQAwADAAMAB%%3D%3D

**The special characters have been replaced by their % equivalents, preventing an URL malformation.**

## Return value

Once executed, the Web Service will return a JSON Response.

The response has the following format:

{

“result”: RETURN\_STATE,

“message”: RETURN\_MESSAGE

}

The return state can be:

* Success
* Error
* Timeout

If the command succeeded, the response will only contain the result, it won’t contain a message.

If the command response indicates an error or a timeout, a message will be provided in the JSON structure to get information on why the command ended up with an error or a timeout.

## Data Encoding

The data that needs to be printed can be passed as plain text or as base64 encoded data.

The choice is made using the encoding parameter which must be set to base64encoded when user wants to use encoded data.

By default, the Web service will process the data with the UTF\_8 character set.

The character set can be specified using the parameter standardCharsets.

It supports following values: UTF\_8, UTF\_16, UTF\_16BE, UTF\_16LE, ISO\_8859\_1, US\_ASCII

The encoding tests were done with the following site:

https://www.base64encode.org/

You can use the following API (MIT license) to encode directly from javascript: https://github.com/emn178/hi-base64

The data **MUST be URI encoded before being used with the encodeURIComponent method** (see URL Format chapter for more information: [URL Format](#_URL_Format))

Examples:

* Print a single line without encoding:   
  http://localhost:8080/?command=printsingleline&text=This is the text to print
* Base64 UTF\_8 encoded data:  
  http://localhost:8080/?command=printsingleline&encoding=base64encoded&standardCharsets=UTF\_8&text=Q2FwdHVyZSB5b3VyIHBlcmZvcm1hbmNlIGVkZ2UNCndpdGggWkVCUkEgU29sdXRpb25zLg%3D%3D
* Base64 UTF-16 encoded data:  
  http://localhost:8080/?command=printzpl&encoding=base64encoded& standardCharsets=UTF\_16&template=%2F%2F5DAFQAfgB%2BAEMARAAsAH4AQwBDAF4AfgBDAFQAfgBeAFgAQQB%2BAFQAQQAwADAAMAB%%3D%3D

## Commands

### Connect

The connect command allows you to connect a printer to the device.

The connection can be made over Bluetooth, Wifi or Ethernet.

If the printer is connected both to Wifi and Bluetooth, the Wifi connection will be privileged to get the best performances.

A device connected to the Wifi can be connected to an ethernet printer if they are on the same network.

The REST command is: connect

The Bluetooth Mac address is specified with the parameter: bluetoothMAC

The Bluetooth Wifi address is specified with the parameter: wifiMAC

The Bluetooth Ethernet address is specified with the parameter: ethernetMAC

Example of connecting to a QLn320 that has Bluetooth and wifi capabilities:

http://localhost:8080/?command=connect&bluetoothMAC=XXXXXXXXXXXX&wifiMAC=XXXXXXXXXXXX

Example of connecting to a ZT600 industrial printer that is connected through ethernet:

http://localhost:8080/?command=connect&ethernetMAC=XXXXXXXXXXXX

### ZPL Printing

This command allows you to send raw ZPL code to the printer.

The REST command is: printzpl

The ZPL data are specified with the parameter: template

Example in plaintext:

http://localhost:8080/?command=printzpl&template= ^XA^FO10,10^ADN,300,70^FDTEST^FS^XZ

Example with base64 encoding:

http://localhost:8080/?command=printzpl&encoding=base64encoded&standardCharsets=UTF\_8&template=XlhBXkZPMTAsMTBeQURBLDE1MCw0MF5GRFRlc3QgWlBMXkZTXlha

### ZPL Printing with variable data

This command allows you to send raw ZPL code with variable parameters to the printer.

The REST command is: printzpl

The ZPL data are specified with the parameter: template

The variable data are specified with the parameter: variables

Variable data must be composed of value/key pairs separated by “:”

The number of parameters separated by “:” must be pair (the reminder of a division by 2 must be null)

As mentioned before, **the template and the variable data MUST be URI Encoded** before creating the full URL.

Example of variable data encoded and not encoded:

* Not URI Encoded variables:  
  %MYVAR1KEY%:myvar1value:%MYVAR2KEY%:myvar2value:%MYVAR3KEY%:myvar3key
* URI Encoded variables:  
  %25MYVAR1KEY%25%3Amyvar1value%3A%25MYVAR2KEY%25%3Amyvar2value%3A%25MYVAR3KEY%25%3Amyvar3key

**If the data are passed as a base64 encoded string, the variables MUST be encoded in base64 as well (the encoding must be the same than the one used for the template data).**

URL Intent Example in plaintext:

http://localhost:8080/?command=printzpl&template=CT~~CD%2C~CC%5E~CT~%5EXA~TA000~JSN%5ELT0%5EMNN%5EMTD%5EPON%5EPMN%5ELH0%2C0%5EJMA%5EPR5%2C5~SD10%5EJUS%5ELRN%5ECI0%5EXZ%5EXA%5EMMT%5EPW406%5ELL0200%5ELS0%5EFT38%2C83%5EA0N%2C27%2C28%5EFH%5C%5EFDPierre%20d'ouverture%2C%20%5EFS%5EFT38%2C116%5EA0N%2C27%2C28%5EFH%5C%5EFDli%5C82e%20au%20%25TEXTE%25%20du%20coeur%20et%20%5EFS%5EFT38%2C149%5EA0N%2C27%2C28%5EFH%5C%5EFD%5C85%20v%5C82nus%20apaisante%2C%20favorise%20%5EFS%5EFT38%2C182%5EA0N%2C27%2C28%5EFH%5C%5EFDla%20gu%5C82rison%20des%20%5C82motions%20%25CODE%25%5EFS%5EFT107%2C44%5EA0N%2C39%2C38%5EFH%5C%5EFDMALACHITE%5EFS%5EFO304%2C29%5EGB79%2C0%2C8%5EFS%5EFO13%2C29%5EGB79%2C0%2C8%5EFS%5EPQ1%2C0%2C1%2CY%5EXZ&variables=%25TEXTE%25%3Amontexte%3A%25CODE%25%3A12345

Example in base64 ASCII Encoded data:

http://localhost:8080/?command=printzpl&encoding=base64encoded&standardCharsets=US\_ASCII&template=Q1R%2BfkNELH5DQ15%2BQ1R%%3D&variables=JVRFWFRFJTptb250ZXh0ZTolQ09ERSU6MTIzNDU%3D

### FileName Print

This command will allow you to print a template file (refer to PrintConnect guide for more information)

The REST command is: printtemplatefile

The filename is passed using the parameter: filename

Optional variable data can be passed using the parameter: variables

Examples:

http://localhost:8080/?command=printtemplatefile&filename=mytemplate.zpl

http://localhost:8080/?command=printtemplatefile&filename=mytemplate.zpl&variables=%25TEXTE%25%3Amontexte%3A%25CODE%25%3A12345

### Line Print

This command allows to send raw text to the printer.

The REST command is: printsingleline

The text to print is passed using the parameter: text

**Raw text MUST be URI Encoded if it contains spaces or special characters like /,=,+,&,…**

**Raw text MUST be base64 encoded in if it contains carriage returns.**

Example of printing a line of text without spaces or special characters:

http://localhost:8080/?command=printsingleline&text=ATextToPrintWithoutSpacesOrSpecialCharacters

Example of printing a line of text that contains spaces and special characters:

http://localhost:8080/?command=printsingleline&text=A%20text%20to%20print%20with%20speci%40l%20%2F%2F\*%2B%2B%3D%3Dch%40r%40cters%3D%3D%2B%2B\*%2F%2F%20and%20spaces

// Print raw text with carriage returns encoded in base64 UTF-16 CRLF

http://localhost:8080/?command=printsingleline&encoding=base64encoded&standardCharsets=UTF\_16&text=%2F%2F5UAGgAaQBzACAAaQBzACAAYQAgAHQAZQB4AHQAIAB3AGkAdABoACAAQwBSAEwARgAgAA0ACgBjAGEAcgByAGkAYQBnAGUAIAByAGUAdAB1AHIAbgBzACAAKAB3AGkAbgBkAG8AdwBzACkADQAKAEEAbgBkACAAcwBvAG0AZQAgAHMAcABlAGMAaQBhAGwADQAKAGMAaABAAHIAQABjAHQAZQByAHMAIABsAGkAawBlADoADQAKACoAKgArACsAPQA9AC8ALwAkACQAXgBeAF4AXgBeAA0ACgB8AEUAbgBkACAAbwBmACAAdAByAGEAbgBzAG0AaQBzAHMAaQBvAG4AfAA%3D

### Passthrough Print

Passthrough print allows to send raw data to the printer.

The following data can be send to the printer:

* ZPL
* CPCL
* SetGetDo commands
* Raw text

These data will be interpreted by the printer according to its settings.

Ex: ZPL text send when the printer is in RawText mode will print un-processed ZPL content instead of printing the interpretation of this content.

Check PrintConnect User Guide for more information on passthrough printing.

The REST command is: passthrough

The text to print is passed using the parameter: data

Example of sending Raw text:

http://localhost:8080/?command=passthrough&data=I'm%20a%20raw%20text

Example of sending ZPL code:

http://localhost:8080/?command=passthrough&data=^XA^FO10,10^ADN,300,70^FDTEST^FS^XZ

### Unselect Printer

This command allows to un-select the current Printer.

Once executed, the user won’t be able to print or send anything until a new printer is selected with PrintConnect (or connected using Tap&Pair).

Example:

http://localhost:8080/?command=unselect

# Browser Intent URL Service

Note: This method is not recommended if you plan to use Chrome browser.

It happened that on some versions, the new registered intent schemes sometime stop working.

This does not happen with Firefox and Opera, and maybe other browsers.

If you plan to use Chrome, and the intent suddenly stop working, it is advised to run Firefox once to fix Chrome behaviour.

## How To

The printer must be paired using PrintConnect for the wrapper to work.

To do this, simply install PrintConnect, and then do an NFC pairing with a mobile bluetooth printer or any other kind of pairing (Wifi, Ethernet)

The process is explained in the PrintConnect user guide.

The commands can be executed using a href in an HTML element or simply by using the window.open("") method;

Example of line-to-line printing using the javascript window.open method

window.open("intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;end");

## URL Format

The commands are passed to the intent as a “virtual server”.

The target package has to be specified in the URL (i.e. the browserintenturl service package).

The URL are formed this way:

Intent://mycommand/#Intent;scheme=myscheme;package=com.zebra.enterpriseservices;mycommandextras;S.verbose=myverbosechoice;end

The text in dark blue represent the mandatory part of the URL.

The text in green represent the variable members of the URL.

## Feedback control

The extra S.verbose can be passed to the URL to control the display of information popups or not.

S.verbose=true will display popup informations.

S.verbose=false will not display popup informations.

## Data Encoding

The data that needs to be printed can be passed as plain text or as base64 encoded data.

The choice is made using the scheme attribute which can take the values: plaintext or base64encoded.

The extra S.standardCharsets specifies the type of the base64 encoded string. It supports following values: UTF\_8, UTF\_16, UTF\_16BE, UTF\_16LE, ISO\_8859\_1, US\_ASCII

The encoding tests were done with the following site:

https://www.base64encode.org/

You can use the following API (MIT license) to encode directly from javascript: https://github.com/emn178/hi-base64

If you use non-encoded strings, it is better to "escape" them before passing them to the URL:

https://www.freeformatter.com/javascript-escape.html

Examples:

* Plain text intent:   
  intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;end
* Base64 ASCII encoded data:  
  intent://printzpl/#Intent;scheme=base64encoded;package=com.zebra.enterpriseservices;S.template=;S.standardCharsets=US\_ASCII;S.verbose=true;end
* Base64 UTF-16 encoded data:  
  intent://printzpl/#Intent;scheme=base64encoded;package=com.zebra.enterpriseservices;S.template=;S.standardCharsets=UTF\_16;S.verbose=true;end

## Quit Parameter

When a print intent is send by Chrome, the application will momentarily open a transparent window on top of Chrome (or any other browser that supports Chrome Intents URL), and will quit.

The Quit parameter allows to control.

The extra S.quitmode can be passed to the URL to control how the application will return the focus to the browser.

The following values can be used:

* FINISH\_AFFINITY // Default value. Will finish current task affinity
* FINISH\_AND\_REMOVE\_TASK // Will finish current task affinity and remove it
* MOVE\_TASK\_TO\_BACK // Will move print task to the back of the app stack
* KILL\_PROCESS // Kill task process
* SYSTEM\_EXIT // Use System.Exit to kill the application
* LAUNCH\_INTENT // Launch an intent to put an app to the front

When the print job is sent using the LAUNCH\_INTENT parameters, **the extra S.component must be provided in the intent URL**:

* S.component  
  The component name of the application we wants to put on front.

Examples:

* Simulate a press back to remove transparent window:   
  intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;S.quitmode=PRESS\_BACK;end
* Kill the application when the print job is done:   
  intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;S.quitmode=SYSTEM\_EXIT;end
* Put the Chrome browser on front when the job print is done:  
  intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;S.quitmode=LAUNCH\_INTENT;S.component= com.android.chrome; end
* Put the Firefox browser on front when the job print is done:  
  intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;S.quitmode=LAUNCH\_INTENT;S.component= org.mozilla.firefox;end

## Commands

### ZPL Printing

This command allows you to send raw ZPL code to the printer.

The URL virtual server is: printzpl

The ZPL data is passed with the extra: S.template

Example:

intent://printzpl/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.template=CT~~CD,~CC^~CT~^XA~TA000~JSN^LT0^MNN^MTD^PON^PMN^LH0,0^JMA^PR5,5~SD10^JUS^LRN^CI0^XZ^XA^MMT^PW406^LL0200^LS0^FT38,83^A0N,27,28^FH\\^FDPierre d\'ouverture, ^FS^FT38,116^A0N,27,28^FH\\^FDli\\82e au chakra du coeur et ^FS^FT38,149^A0N,27,28^FH\\^FD\\85 v\\82nus apaisante, favorise ^FS^FT38,182^A0N,27,28^FH\\^FDla gu\\82rison des \\82motions^FS^FT107,44^A0N,39,38^FH\\^FDMALACHITE^FS^FO304,29^GB79,0,8^FS^FO13,29^GB79,0,8^FS^PQ1,0,1,Y^XZ;S.verbose=true;end

### ZPL Printing with variable data

This command allows you to send raw ZPL code with variable parameters to the printer.

The URL virtual server is: printzpl

The ZPL data is passed with the extra: S.template

The variable data are passed with the extra: S.variables

Variable data must be composed of value/key pairs separated by “:”

The number of parameters separated by “:” must be pair (the reminder of a division by 2 must be null)

Example of variable data extra:

S.variables=%MYVAR1KEY%:myvar1value:%MYVAR2KEY%:myvar2value:%MYVAR3KEY%:myvar3key

**If the data are passed as a base64 encoded string, the variables MUST be encoded with the same format.**

URL Intent Example:

intent://printzpl/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.template=CT~~CD,~CC^~CT~^XA~TA000~JSN^LT0^MNN^MTD^PON^PMN^LH0,0^JMA^PR5,5~SD10^JUS^LRN^CI0^XZ^XA^MMT^PW406^LL0200^LS0^FT38,83^A0N,27,28^FH\\^FDPierre d\'ouverture, ^FS^FT38,116^A0N,27,28^FH\\^FDli\\82e au %TEXTE% du coeur et ^FS^FT38,149^A0N,27,28^FH\\^FD\\85 v\\82nus apaisante, favorise ^FS^FT38,182^A0N,27,28^FH\\^FDla gu\\82rison des \\82motions %CODE%^FS^FT107,44^A0N,39,38^FH\\^FDMALACHITE^FS^FO304,29^GB79,0,8^FS^FO13,29^GB79,0,8^FS^PQ1,0,1,Y^XZ;S.variables=%TEXTE%:montexte:%CODE%:12345;S.verbose=true;end

ASCII Encoded data:

intent://printzpl/#Intent;scheme=base64encoded;package=com.zebra.enterpriseservices;S.template=;S.variables=JVRFWFRFJTptb250ZXh0ZTolQ09ERSU6MTIzNDU=;S.standardCharsets=US\_ASCII;S.verbose=true;end

### FileName Print

This command will allow you to print a template file (refer to PrintConnect guide for more information)

The URL virual server is: printtemplatefile

The filename is passed using the extra: S.filename

Optional variable data can be passed using the extra: S.variables

Examples:

intent://printtemplatefile/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.filename=mytemplate.zpl;S.verbose=true;end

intent://printtemplatefile/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.filename=mytemplate.zpl;S.variables=%TEXTE%:montexte:%CODE%:12345;S.verbose=true;end

### Line Print

This command allows to send raw text to the printer.

The URL virual server is: printsingleline

The text to print is passed using the extra: S.text

Raw text MUST be encoded in base64 if it contains carriage returns.

Examples:

intent://printsingleline/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.text=This is the text to print;S.verbose=true;end

// Print raw text with carriage returns encoded in base64 UTF-16 CRLF

intent://printsingleline/#Intent;scheme=base64encoded;package=com.zebra.enterpriseservices;S.text=//5UAGgAaQBzACAAaQBzACAAbQB5ACAAdABlAHgAdAANAAoAVABvACAAcAByAGkAbgB0AA0ACgBFAG4AYwBvAGQAZQBkACAAaQBuACAAYgBhAHMAZQA2ADQADQAKAFcAaQB0AGgAIABjAGEAcgByAGkAYQBnAGUAIAByAGUAdAB1AHIAbgBzAA0ACgANAAoA;S.standardCharsets=UTF\_16;S.verbose=true;end

### Passthrough Print

Passthrough print allows to send raw data to the printer.

These data will be interpreted by the printer according to its current mode of processing.

The following data can be send to the printer:

* ZPL
* CPCL
* SetGetDo commands
* Raw text

Read PrintConnect user guide for more information on passthrough printing.

The URL virual server is: passthrough

The text to print is passed using the extra: S.data

Example:

intent://passthrough/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.data=CT~~CD,~CC^~CT~^XA~TA000~JSN^LT0^MNW^MTD^PON^PMN^LH0,0^JMA^PR5,5~SD10^JUS^LRN^CI0^XZ^XA^MMT^PW406^LL0200^LS0^FT38,83^A0N,27,28^FH\\^FDPierre d\'ouverture, ^FS^FT38,116^A0N,27,28^FH\\^FDli\\82e au chakra du coeur et ^FS^FT38,149^A0N,27,28^FH\\^FD\\85 v\\82nus apaisante, favorise ^FS^FT38,182^A0N,27,28^FH\\^FDla gu\\82rison des \\82motions^FS^FT107,44^A0N,39,38^FH\\^FDMALACHITE^FS^FO304,29^GB79,0,8^FS^FO13,29^GB79,0,8^FS^PQ1,0,1,Y^XZ;S.verbose=true;end

### Unselect Printer

This command allows to un-select the current Printer.

Once executed, the user won’t be able to print or send anything until a new printer is selected with PrintConnect (or connected using Tap&Pair).

Example:

intent://unselect/#Intent;scheme=plaintext;package=com.zebra.enterpriseservices;S.verbose=true;end

# Appendix

## Licence

**END USER LICENSE AGREEMENT (UNRESTRICTED SOFTWARE)**

IMPORTANT PLEASE READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Zebra International Holdings Corporation ("Zebra") for software, owned by Zebra and its affiliated companies and its third party suppliers and licensors, that accompanies this EULA. ("Software"). BY USING THE SOFTWARE, YOU ACKNOWLEDGE ACCEPTANCE OF THE TERMS OF THIS EULA. IF YOU DO NOT ACCEPT THESE TERMS, DO NOT USE THE SOFTWARE.

1. GRANT OF LICENSE. Zebra grants you, End-User Customer, the following rights provided that you comply with all terms and conditions of this EULA: For Software associated with Zebra hardware, Zebra hereby grants you ("Licensee" or "you") a personal, nonexclusive, nontransferable, nonassignable, nonsublicenseable license to use the Software subject to the terms and conditions of this Agreement. Only your employees or subcontractors may use the Software. You shall take all necessary steps to insure that your employees and subcontractors abide by the terms of this Agreement. You shall use the Software only for your internal business purposes, exclusively to support the Zebra hardware, including the right to (i) use, modify, and incorporate all or portions of the sample source code (the "Sample Code"), runtime library files, and/or documentation files that may be included in the unmodified Software into Licensees own programs (the "User Programs") to support the Zebra hardware exclusively, provided that no license is granted herein under any patents that may be infringed by Licensees modifications, derivative works or by other works in which any portion of the Software may be incorporated; (ii) distribute the Sample Code in object code format only as substantially modified or only as part of the User Programs to support the Zebra hardware exclusively; and (iii) distribute the runtime library files in their original form to support the Zebra hardware exclusively. For a standalone Software application, you may install, use, modify, and incorporate all or portions of any object code, available source code ("Source Code"), runtime library files, and/or documentation files that may be included with the unmodified Software into your own programs (the "User Programs") and distribute the User Programs to third parties. Any use of the Software outside of the conditions set forth herein is strictly prohibited and will be deemed a breach of this Agreement resulting in immediate termination of your License. Zebra will be entitled to all available remedies at law or in equity (including immediate injunctive relief and repossession of all Software unless Licensee is a Federal agency of the United States Government).

Certain items of the Software may be subject to open source licenses. The open source license provisions may override some of the terms of this EULA. Zebra makes the applicable open source licenses available to you on a Legal Notices readme file available on your device and/or in System Reference guides or in Command Line Interface (CLI) Reference guides associated with certain Zebra products.

1. RESERVATION OF RIGHTS AND OWNERSHIP. Zebra reserves all rights not expressly granted to you in this EULA. The Software is protected by copyright and other intellectual property laws and treaties. Zebra or its suppliers own the title, copyright and other intellectual property rights in the Software. The Software is licensed, not sold.
2. LIMITATIONS ON END USER RIGHTS. You shall not distribute, sublicense, rent, loan, lease, export, re-export, resell, ship or divert or cause to be exported, re-exported, resold, shipped or diverted, directly or indirectly, the unmodified Software under this Agreement. You shall not, and shall not permit others to: (i) combine the Software including any Sample or Source Code, in whole or in part, with any Open Source Software having license terms and obligations that include copyleft obligations and/or intellectual property encumbrances; (ii) remove any proprietary notices, marks, labels, or logos from the Software; (iii) rent or transfer all or some of the Software to any other party without Zebras prior written consent; or (iv) utilize any computer software or hardware which is designed to defeat any copy protection device, should the Software be equipped with such a protection device.
3. MACHINE DATA. Machine Data means anonymized usage data collected by devices sold (or licensed) under this Agreement such as battery management (time to empty, standby current, average current), device system time, CPU processing load, free RAM, number of running processes, network information (name, identifier), device identifier, firmware version, hardware version device type, audio volume, LED state, beeper volume, backlight level, key light, odometer count, reboot, reboot cause, total storage and physical memory availability, power cycle count, and device up time. Notwithstanding anything else in this Agreement, all title and ownership rights in and to Machine Data are held by Zebra. In the event, and to the extent you are deemed to have any ownership rights in Machine Data, you hereby grant Zebra a limited, revocable, non-exclusive right and license to use Machine Data.
4. LOCATION INFORMATION. The Software may enable you to collect location-based data from one or more client devices which may allow you to track the actual location of those client devices. Zebra specifically disclaims any liability for your use or misuse of the location-based data. You agree to pay all reasonable costs and expenses of Zebra arising from or related to third party claims resulting from your use of the location-based data.
5. SOFTWARE RELEASES. Zebra may periodically release new versions of the Software which will be made available to you.
6. EXPORT RESTRICTIONS. You acknowledge that the Software is subject to export restrictions of various countries. You agree to comply with all applicable international and national laws that apply to the Software, including all the applicable export restriction laws and regulations.
7. ASSIGNMENT.You may not assign this Agreement or any of your rights or obligations hereunder (by operation of law or otherwise) without the prior written consent of Zebra. Zebra may assign this Agreement and its rights and obligations without your consent. Subject to the foregoing, this Agreement shall be binding upon and inure to the benefit of the parties to it and their respective legal representatives, successors and permitted assigns.
8. TERMINATION. This EULA is effective until terminated. Your rights under this License will terminate automatically without notice from Zebra if you fail to comply with any of the terms and conditions of this EULA. Zebra may terminate this Agreement by offering you a superseding Agreement for the Software or for any new release of the Software and conditioning your continued use of the Software or such new release on your acceptance of such superseding Agreement. Upon termination of this EULA, you must cease all use of the Software and destroy all copies, full or partial, of the Software.
9. DISCLAIMER OF WARRANTY. UNLESS SEPARATELY STATED IN A WRITTEN EXPRESS LIMITED WARRANTY, ALL SOFTWARE PROVIDED BY ZEBRA IS PROVIDED "AS IS" AND ON AN "AS AVAILABLE" BASIS, WITHOUT WARRANTIES OF ANY KIND FROM ZEBRA, EITHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT POSSIBLE PURSUANT TO APPLICABLE LAW, ZEBRA DISCLAIMS ALL WARRANTIES EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY OR WORKMANLIKE EFFORT, FITNESS FOR A PARTICULAR PURPOSE, RELIABILITY OR AVAILABILITY, ACCURACY, LACK OF VIRUSES, NON INFRINGEMENT OF THIRD PARTY RIGHTS OR OTHER VIOLATION OF RIGHTS. ZEBRA DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE. TO THE EXTENT THAT THE SOFTWARE COVERED BY THIS EULA INCLUDES EMULATION LIBRARIES, SUCH EMULATION LIBRARIES DO NOT WORK 100% CORRECTLY OR COVER 100% OF THE FUNCTIONALITY BEING EMULATED, ARE OFFERED "AS IS" AND WITH ALL FAULTS, AND ALL THE DISCLAIMERS AND LIMITATIONS CONTAINED IN THIS PARAGRAPH AND THIS AGREEMENT APPLY TO SUCH EMULATION LIBRARIES.SOME JURISDICTIONS DO NOT ALLOW EXCLUSIONS OR LIMITATIONS OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS OR LIMITATIONS MAY NOT APPLY TO YOU. NO ADVICE OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED BY YOU FROM ZEBRA OR ITS AFFILIATES SHALL BE DEEMED TO ALTER THIS DISCLAIMER BY ZEBRA OF WARRANTY REGARDING THE SOFTWARE, OR TO CREATE ANY WARRANTY OF ANY SORT FROM ZEBRA.
10. THIRD-PARTY APPLICATIONS. Certain third party applications may be included with, or downloaded with this Software. Zebra makes no representations whatsoever about any of these applications. Since Zebra has no control over such applications, you acknowledge and agree that Zebra is not responsible for such applications. You expressly acknowledge and agree that use of third party applications is at your sole risk and that the entire risk of unsatisfactory quality, performance, accuracy and effort is with you. You agree that Zebra shall not be responsible or liable, directly or indirectly, for any damage or loss, including but not limited to any damage to or loss of data, caused or alleged to be caused by, or in connection with, use of or reliance on any such third party content, products, or services available on or through any such application. You acknowledge and agree that the use of any third-party application is governed by such third party application provider's Terms of Use, License Agreement, Privacy Policy, or other such agreement and that any information or personal data you provide, whether knowingly or unknowingly, to such third-party application provider, will be subject to such third party application provider's privacy policy, if such a policy exists. ZEBRA DISCLAIMS ANY RESPONSIBILITY FOR ANY DISCLOSURE OF INFORMATION OR ANY OTHER PRACTICES OF ANY THIRD PARTY APPLICATION PROVIDER. ZEBRA EXPRESSLY DISCLAIMS ANY WARRANTY REGARDING WHETHER YOUR PERSONAL INFORMATION IS CAPTURED BY ANY THIRD PARTY APPLICATION PROVIDER OR THE USE TO WHICH SUCH PERSONAL INFORMATION MAY BE PUT BY SUCH THIRD PARTY APPLICATION PROVIDER.
11. LIMITATION OF LIABILITY. ZEBRA WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR RELATING TO THE USE OR THE INABILITY TO USE THE SOFTWARE OR ANY THIRD PARTY APPLICATION, ITS CONTENT OR FUNCTIONALITY, INCLUDING BUT NOT LIMITED TO DAMAGES CAUSED BY OR RELATED TO ERRORS, OMISSIONS, INTERRUPTIONS, DEFECTS, DELAY IN OPERATION OR TRANSMISSION, COMPUTER VIRUS, FAILURE TO CONNECT, NETWORK CHARGES, IN-APP PURCHASES, AND ALL OTHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES EVEN IF ZEBRA 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 EXCLUSIONS OR LIMITATIONS MAY NOT APPLY TO YOU. NOTWITHSTANDING THE FOREGOING, ZEBRA’S TOTAL LIABILITY TO YOU FOR ALL LOSSES, DAMAGES, CAUSES OF ACTION, INCLUDING BUT NOT LIMITED TO THOSE BASED ON CONTRACT, TORT, OR OTHERWISE, ARISING OUT OF YOUR USE OF THE SOFTWARE OR THIRD PARTY APPLICATIONS, OR ANY OTHER PROVISION OF THIS EULA, SHALL NOT EXCEED THE FAIR MARKET VALUE OF THE SOFTWARE OR AMOUNT PURCHASER PAID SPECIFICALLY FOR THE SOFTWARE. THE FOREGOING LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS (INCLUDING SECTIONS 10, 11, 12, AND 15) SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.
12. INJUNCTIVE RELIEF.You acknowledge that, in the event you breach any provision of this Agreement, Zebra will not have an adequate remedy in money or damages. Zebra shall therefore be entitled to obtain an injunction against such breach from any court of competent jurisdiction immediately upon request without posting bond. Zebra's right to obtain injunctive relief shall not limit its right to seek further remedies.
13. MODIFICATION.No modification of this Agreement shall be binding unless it is in writing and is signed by an authorized representative of the party against whom enforcement of the modification is sought.
14. U.S. GOVERNMENT END USERS RESTRICTED RIGHTS. This provision only applies to U.S. Government end users. The Software is a commercial item as that term is defined at 48 C.F.R. Part 2.101, consisting of commercial computer software and computer software documentation as such terms are defined in 48 C.F.R. Part 252.227-7014(a)(1) and 48 C.F.R. Part 252.227-7014(a)(5), and used in 48 C.F.R. Part 12.212 and 48 C.F.R. Part 227.7202, as applicable. Consistent with 48 C.F.R. Part 12.212, 48 C.F.R. Part 252.227-7015, 48 C.F.R. Part 227.7202-1 through 227.7202-4, 48 C.F.R. Part 52.227-19, and other relevant sections of the Code of Federal Regulations, as applicable, the Software is distributed and licensed to U.S. Government end users (a) only as a commercial item, and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions contained herein.
15. APPLICABLE LAW. This EULA is governed by the laws of the state of Illinois, without regard to its conflict of law provisions. This EULA shall not be governed by the UN Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded.

# Source code

The source code of the Browser URL Intent Service is available on GitHub:

<https://github.com/ltrudu/PrintConnectSample>