



Scan to Print

[v1.6 Nov 22, 2019]

Update: May 4, 2023 (v1.7)

From

```
b64encoded = replaceAll(b64encoded, "=", "");
```

To:

```
b64encoded = replaceAll(b64encoded, "=", "_");
```

ZPL Example

The first implementation to go live used content from the response text field of a validation database that included the variable data to print. Those variables are embedded in Zebra's ZPL code which controls what's printed and how it's printed.

The variable data are included in the response text. The scan of the barcode value in column 1 of the validation database will show the app-user those variables and present a PRINT key.

1 - Structure of Validation Database

The codeREADr service must be a [Validate Scans](#) service type. For the database you associate to that service here is the required structure:

Column 1 is the barcode value.

Column 2 includes structured text based on variables in the ZPL code. For example, if there are three variables ...

`_comp_price_`

`_our_price_`

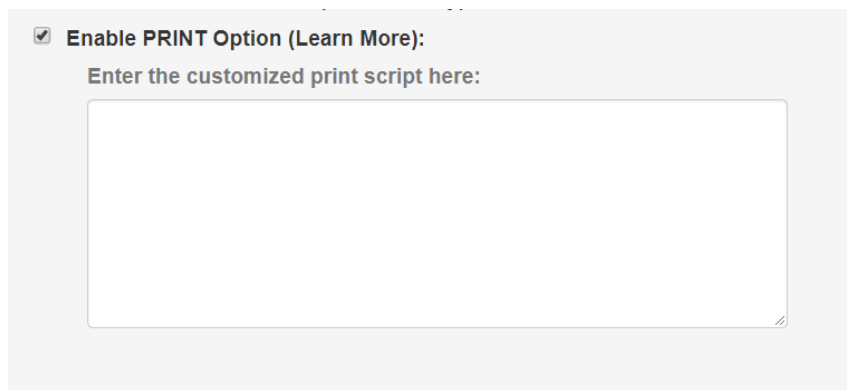
`_barcode_`

... then the structured text would look like this ...

```
{"_comp_price_":"110.50","_our_price_":"100.99","_barcode_":"9038748"}
```

2 - Adding Script and ZPL Code

One the Advanced step when creating/editing your codeREADr service you check "Enable PRINT Option".



Enable PRINT Option (Learn More):
Enter the customized print script here:

Below you will then find a form field to paste the print script.

3 - Example Script

Notes:

a) You need this **whole print script** pasted into the form field from `<html>` to `</html>` but you must replace the example ZPL code with your own ZPL code.

b) The ZPL code has the background highlighted in grey. Changing that code changes what's printed.

```
<html><head>
```

```
  <script type="text/codereadr" id="rawCode">
```

```
  ^PCT~~CD,~CC^~CT~
```

```
  ^XA~TA000~JSN^LT0^MNM^MTD^PON^PMN^LH0,0^JMA^PR1,1~SD25^JUS^LRN^CI
```

```
0^XZ
^XA
^MMT
^PW384
^LL0203
^LS0
^FT120,176^A0I,23,24^FH^FDOur Price^FS
^FT384,175^A0I,23,24^FH^FDCompare Price ^FS
^FT371,122^A0I,47,43^FH^FD$_comp_price_^FS
^FT133,123^A0I,47,43^FH^FD$_our_price_^FS
^BY2,2,69^FT376,40^B3I,N,,N,N
^FD_barcode_^FS
^FT283,13^A0I,28,28^FB176,1,0,C^FH^FD_barcode_^FS
^PQ1,0,1,Y^XZ
</script>
```

```
<script type='text/javascript'>
  var rawCode = document.getElementById("rawCode").innerHTML.trim();

  function escapeRegExp(str) {
    return str.replace(/([.*+?^=!:${}()|\[\]\/\\])/g, "\\$1");
  }

  function replaceAll(str, find, replace) {
    return str.replace(new RegExp(escapeRegExp(find), 'g'), replace);
  }

  function getPrintVars(crData) {
    try {
      return JSON.parse(crData.scanResponse);
    } catch (err) {
      console.log("Trouble: "+err);
      return {};
    }
  }

  function getPrintCallback(crData) {
    var printVars = getPrintVars(crData);
    var zplCode = rawCode;
```

```

for (var key in printVars) {
  if (printVars.hasOwnProperty(key)) {
    zplCode = replaceAll(zplCode, key, printVars[key]);
  }
}

var b64encoded = btoa(unescape(encodeURIComponent(zplCode)));

b64encoded = replaceAll(b64encoded, "=", "_");
return {"url":"arrowhead://x-callback-url/zplcode?code="+b64encoded};
}
</script>

```

```
</head><body></body></html>
```

4 - Variable Data Available to Print

The "crData" variable in the PRINT javascript above is set to "crData.scanResponse" which will return the "result" (response) after a scan.

Here are some of the other variable options:

```

{
  "scanValue":"050428375624",
  "scanStatus":"1",
  "scanResponse":"Recorded to device.",
  "answers":[
    {
      "qtype":"manual",
      "value":["Donated $5 to the cause." ],
      "qtext":"Comments:",
      "eid": "",
      "qid":"1525"
    },
    {
      "qtype":"checkbox",
      "value":["yellow", "green" ],

```

```

"qtext":"Colors",
"eid":"","
"qid":"303"
}
],
"properties":{
"prop1":"value1",
"time_zone":"GMT-04:00, GMT-4",
"capture_type":"camera_scan",
"prop2":"value2",
"app_used":"2.11.14r1625 for iOS-10.2.1",
"barcode_format":"UPC_A",
"scanned_at_utc":"2017-06-02T23:32:00.260Z",
"scan_engine":"qmv1.9.6r4502"
},
"context":{
"username":"MyUser",
"user_id":"39430",
"service_id":"9859",
"service":"MyService",
"device_id":"39434"
}
}
}

```

5 - Printing the Answers to Questions

Here's an example for printing the answers to Session Questions and Pre-Submit Questions:

....

```

function getAnswerString(crData, qid, separator) {
    if (Array.isArray(crData.answers) && crData.answers.length > 0) {
        var answerObject = false;
        if (qid === false) {
            answerObject = crData.answers[0];
        } else {

```

```
        answerObject = crData.answers.find(a => a.qid == qid);
    }
    if (answerObject) {
        return answerObject.value.join(separator ? separator : "");
    }
}
return "";
}
```

```
function getPrintVars(crData) {
    try {
        return {
            "_barcode_":crData.scanValue,
            "_our_price_":crData.scanResponse,
            "_comp_price_":getAnswerString(crData,"4937")
        };
    } catch (err) {
        console.log("Trouble: "+err);
    }
    return {};
}
```