



Scan to Print Response Text

Steps

These

1. Create a [Validate Scans](#) service type. [Note: You can also use your own servers if you use our Postback URL webhook service type.]
2. On the Advanced Step, check "Enable PRINT Option".
3. Paste your HTML with ZPL code in the form field (see below for script instructions).
1. Upload your [database](#) CSV (manual, [CodeREADr Connector](#), or API). The format needs to be as follows.

Barcode Value (Col 1): The value encoded in the barcodes to be scanned.

Response (Col 2): The associated data to be encoded in the printed label.

Example UPC to GS1 Digital Link:

Barcode UPC: 046100002759

Response: <https://dlink.is/01/00046100002759>

Example GTIN to GS1 Digital Link:

Barcode GTIN: 00046100002759

Response: <https://dlink.is/01/00046100002759>

Example GTIN AI format correction:

Barcode GTIN: (01)00614141999996

Response: (01)00614141999996(10)BATCH1234(17)271231(3102)001452

How to format your PRINT HTML

These instructions are for printing the **Response Text** in the service database.

1. Create your ZPL code. Most AI agents can use the ZPL script shown below (from ^XA to ^XZ) to create the label you want. You should ideally use ZPL to configure label size, print media, and to format the barcode(s) and the content to be printed.
2. Replace the ZPL code from **^XA to ^XZ** in the script below. Keep the rest of the script intact. For custom variations, contact support@codereadr.com.
3. Paste the resulting HTML script on the service's Advanced step's Enable PRINT Option.

HTML

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

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

```

^XA
^PW384
^LL200

; --- QR Code Configuration ---
; Encodes the exact full string from crData.scanResponse
^FO20,20
^BQN,2,4
^FDLA,[FULL_SCAN]^FS

; --- Text Fields ---
^FO170,20
^A0N,26,26
^FDGTIN:^FS

; Dynamic 14-digit GTIN value placed here
^FO170,52
^A0N,26,26
^FD[GTIN_VAL]^FS

^FO170,138
^A0N,34,34
^FDScan Me^FS

^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 getPrintCallback(crData) {
    var zplCode = rawCode;
    var scanStr = crData.scanResponse || "";

    // 1. Extract the 14-digit GTIN value following '/01/'

```

```
var gtinMatch = scanStr.match(/V01V(\d{14})/);
var gtinValue = gtinMatch ? gtinMatch[1] : "00000000000000"; // Fallback if not found

// 2. Inject dynamic values into the ZPL placeholders
zplCode = replaceAll(zplCode, "[FULL_SCAN]", scanStr);
zplCode = replaceAll(zplCode, "[GTIN_VAL]", gtinValue);

// 3. Base64 encode for Arrowhead callback protocol
var b64encoded = btoa(unescape(encodeURIComponent(zplCode)));
b64encoded = replaceAll(b64encoded, "=", "_");

return {"url": "arrowhead://x-callback-url/zplcode?code="+b64encoded};
}
</script>
</head><body></body></html>
```