

Change the Scanned Barcode Value and Response using Multiple Regex Patterns and Replacements.

Alter scanned data to:

1. Required, end-point data formats
2. Normalize data for in-app or external processing
3. Provide value-specific scan responses.

You need to first configure an On-Device Record Scans or On-Device Validate Scans service type. On the Advanced step you need to [copy and paste this script](#) in the form field "Enable on-device custom validation".

Miscellaneous:

☐ Collect Device Location without interruption as a property with each scan

Note: configure Location collection as a Question if the app-user should review the location with each scan.

☐ Enable custom web view 'Lookup'

☐ Disable Scan 'History'

☒ Disable App 'Settings'

☐ Enable custom web view 'History'

☐ Enable custom display configuration

☒ Enable on-device custom validation

Enter your custom validation Javascript ([Learn More](#))

It is executed before each on-device scan is recorded.

```
<script>
var overrideDuplicateChecking = false;
/* Set a error message to block the submit if no match is found. */
var blockSubmitMessage = null;
/* Use regular expressions matches against the scan value to
a) alter scan value with regex group substitutions,
b) replace scan response status and/or message and their response messages.
This is the structure:
[
  new RegExp("SCAN_VAL_PATTERN"), // scan value regex
  "REGEX_REPLACEMENT", // Set to null to keep original
  STATUS, // Set to 0,1,-1 or to null to keep original
  "MESSAGE\n__ORIGINAL_MESSAGE__" // Replace the response text or leave
null to keep original
]
[
  Add this regex and desired response at the end of the list if you want
to override the default response of all values you don't have a specific
pattern to handle.
  Optionally can also include key phrase __ORIGINAL_MESSAGE__ in your custom
```

☐ Enable PRINT Option ([Learn More](#)):

IMPORTANT:

In the script, you need to remove the existing regex patterns and replacements with the ones best suited for your application. The standard script includes 4 separate patterns as examples for normalizing barcodes that don't meet the GS1 GTIN standard. With these patterns, the app will change a 12-digit UPC-A code; an EAN-13 code, an EAN-8, and a 14-digit GTIN with a "01" suffix to GS1-compliant 14-digit GTINs.

```
var regex = [  
    [new RegExp("^(\\d{12})$"), "00$1"],  
    [new RegExp("^(\\d{13})$"), "0$1"],  
    [new RegExp("^(\\d{8})$"), "000000$1"],  
    [new RegExp("^(01(\\d{14})$"), "$1"]  
];
```

Example to change the status (invalid to valid) add a "1" for the expression:

```
[new RegExp("^(\\d{12})$"), "00$1", 1 ],
```

Example to change the response text (invalid to valid) add "Changed to 14-digit GS1 GTIN." in the script.

```
[new RegExp("^(\\d{12})$"), "00$1", 1, "Changed to 14-digit GS1 GTIN."],
```

#