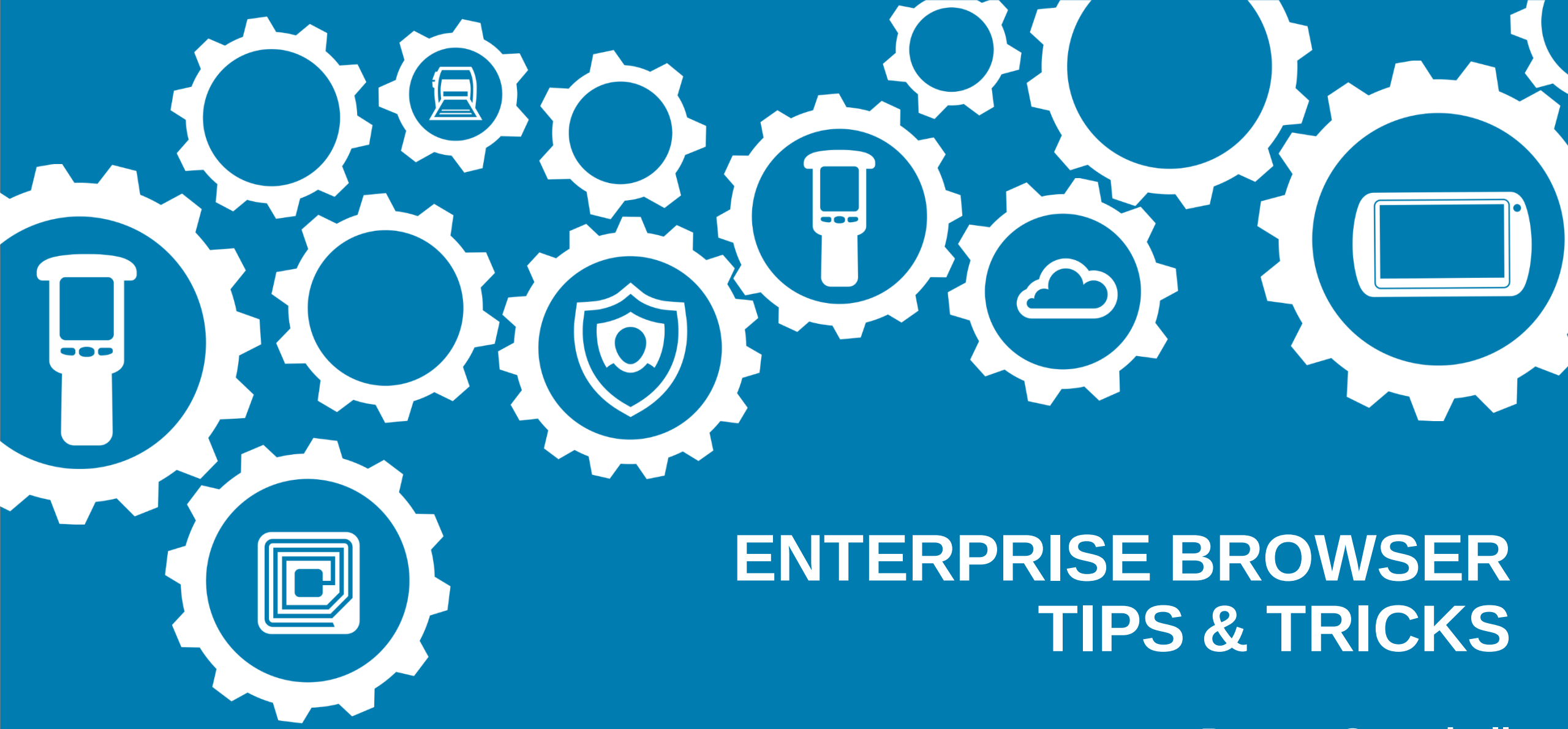
ZEBRA

AMERICAS

APPFORUM 2017

INNOVATE. ENGAGE. TRANSFORM.



ENTERPRISE BROWSER TIPS & TRICKS

Darryn Campbell

Senior Software Architect

Developer Session

Preface

EB Promo During Rho EOS Transition

- Rho is End of Sale **June 30th 2017**
- **Promotion**: Migrate to Enterprise Browser SW license free of charge with purchase of Enterprise Browser Software service support contract

Agenda

EB Tips & Tricks: A customer journey from PocketBrowser to Enterprise Browser

- The existing PocketBrowser application
- Moving to Android with Enterprise Browser
- DOM injection: Improving the look and feel
- DOM injection: Key remapping on a touchscreen device
- DataWedge integration with Enterprise Browser
- Controlling DataWedge from Enterprise Browser
- SimulScan integration with Enterprise Browser
- Enterprise Keyboard Integration

ZEBRA

ZEBRA

APPFORUM 2017
INNOVATE. ENGAGE. TRANSFORM.

OVERVIEW OF APPLICATION

Overview of Application

History

- Application was originally designed for PocketBrowser to scan barcodes and submit to a backend server
  - Scanning was done through DataWedge
  - PocketBrowser APIs were used for Key Capture to facilitate quick data entry

- Requirement: Application must not be changed.
- Requirement: Application must run on new Zebra Android devices with touch screens

Overview of Application

Application working on MC3190 CE6

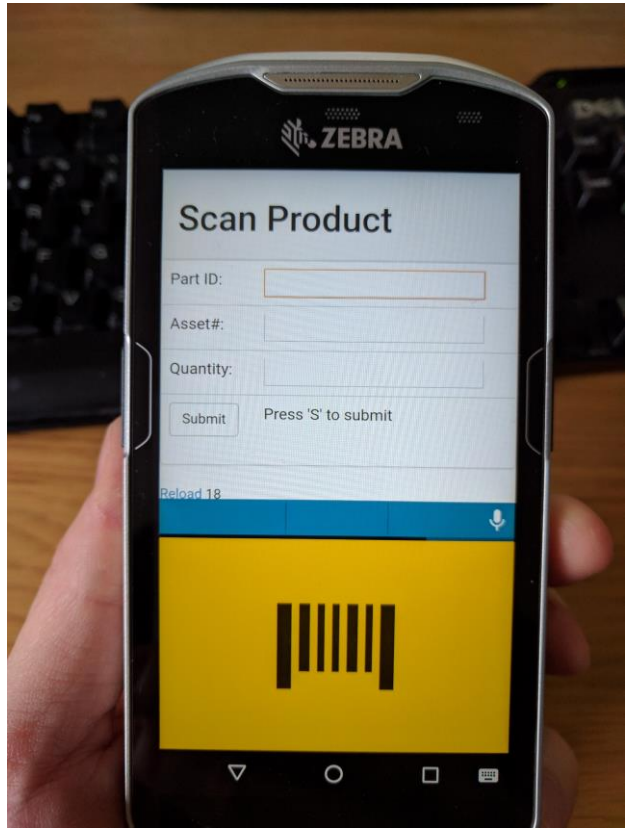
DEMO

Overview of Application

Goal

- Want a visually appealing application which works well on a touch screen and makes employees as efficient as possible

This Presentation

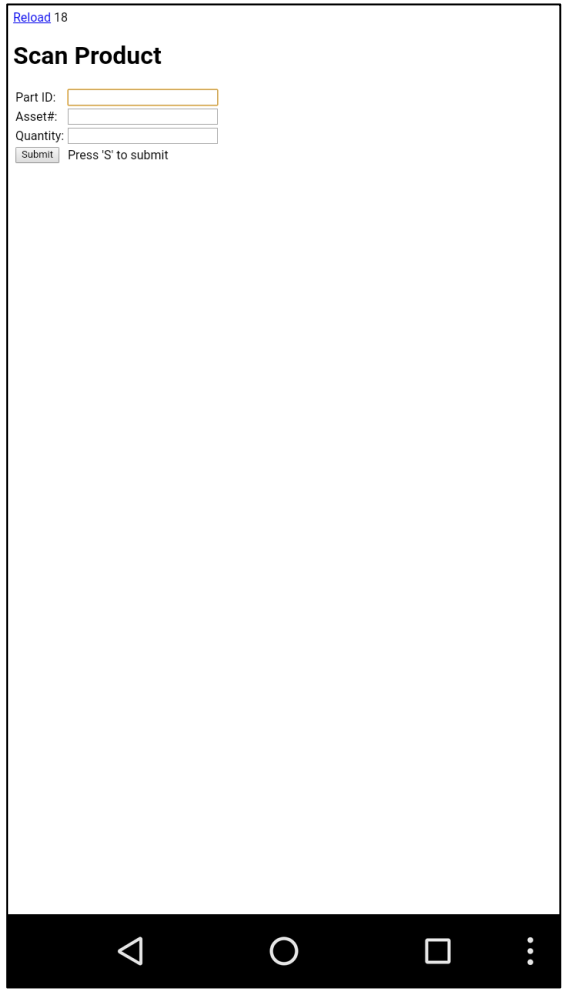



ZEBRA
APPFORUM 2017
INNOVATE. ENGAGE. TRANSFORM.
MODIFYING THE APPLICATION

Modifying the Application

Just run the existing application on an Android device

- It will work, but there are several problems:
  - Without a <Viewport> tag, the page is tiny
  - The page looks just as ugly as it did on CE
  - There is no hardware keyboard, the ‘S’ shortcut key will not work
  - Scanning is still done through DataWedge, received as keystrokes but we can do better
  - The form does not make use of input types since these were not recognized in Pocket IE. We can fix that.



Modifying the Application

Principle: DOM Injection

- Enterprise Browser allows us to inject HTML, CSS and Meta Tags onto the page after it has loaded (http://techdocs.zebra.com/enterprise-browser/1-6/guide/DOMinjection/)
- Config.xml:

```
<CustomDOMElements value="file://%INSTALLDIR%/filelist.txt"/>
```

- Filelist.txt:

```
<script src="./android/dom_inject/bootstrap-3.3.7-dist/js/jquery-3.2.0.js"
pages="*" />
<script src="./android/dom_inject/bootstrap-3.3.7-dist/js/bootstrap.js"
pages="*" />
<link rel="stylesheet" type="text/css" href="./android/dom_inject/bootstrap-
3.3.7-dist/css/bootstrap.min.css" pages="*"/>
```

ZEBRA

Modifying the Application

Adding Bootstrap

- DEMO
- Add ViewPort tag to pages:

```
var viewPortTag=document.createElement('meta');
viewPortTag.id="viewport";
viewPortTag.name = "viewport";
viewPortTag.content = "width=device-width, initial-scale=1";
document.getElementsByTagName('head')[0].appendChild(viewPortTag);
```

- Add attributes to elements so Bootstrap renders them properly:
  - Add panel divs, button attributes, labels for headings etc.

ZEBRA

Modifying the Application

Principle: Calling the EB API

- Any hardware keys we previously used in our application make no sense on a touch screen!
  - We could remap these on the terminal with the KeyMap manager but this is a presentation about Enterprise Browser.
- Again, make use of DOM injection to demonstrate that the Enterprise Browser API set can be used.
  - Need to inject ebapi-modules.js as well as a separate js file to invoke the API
  - EB namespace is now available

ZEBRA

Modifying the Application

Demo: Key Remapping

- DEMO
- New JavaScript calls injected into the DOM:
  - Ebapi-modules.js
  - `EB.KeyCapture.remapKey('0x19', '0x53')`
- This remaps the volume down key on our TC51 to an ‘S’ which is recognized by our existing application.
- Note: EB limitation when injecting DOM from server, order of injection cannot be guaranteed therefore do not call until EB namespace exists.

ZEBRA

Modifying the Application

Principle: Using DataWedge within Enterprise Browser

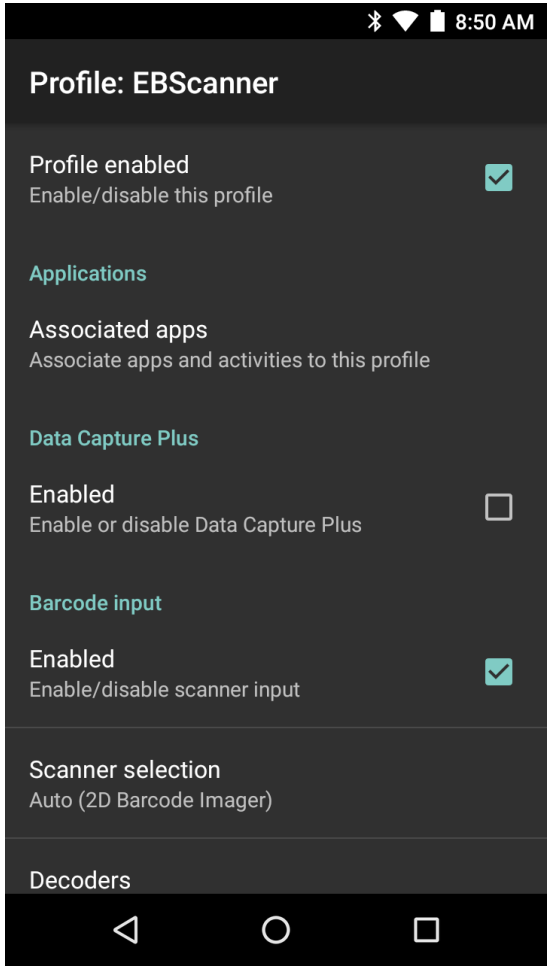
- What is DataWedge?
- Out of the box DataWedge will not work automatically with Enterprise Browser
- There are a few steps to perform:
  - Remove EB profile from DW as explained here: http://techdocs.zebra.com/enterprise-browser/1-6/guide/datawedge/
  - Set **`<UseDWForScanning value="1">`**. By default it is set to 0
  - Configure DataWedge on your device
    - To map exactly with our PocketBrowser application we should send data as keystrokes
- There are several downsides to sending data as keystrokes:
  - Scanner is enabled regardless of which page is shown
  - Text field MUST have focus to receive scanned data

ZEBRA

Modifying the Application

Demo: Using DataWedge within Enterprise Browser

- Demo
- Define profile that comes into effect when Enterprise Browser is visible
- This works the same as our application did under PocketBrowser on CE6 using DataWedge for CE.



ZEBRA

Modifying the Application

Principle: Making better use of DataWedge on Android

- DataWedge on Android can send data via Intents and we can receive these Intents within EB
- EB Configuration:

```
<IntentReceiver>

  <EnableReceiver value="1"/>

  <IntentAction value="com.zebra.ebprofilescan.ACTION"/>

  <IntentCategory value=""/>

</IntentReceiver>
```

- DataWedge exposes an Intent based API. We can use this to control various aspects of DataWedge such as the currently enabled profile

Modifying the Application

Demo: Receiving DataWedge Intents and Controlling DataWedge through the API

- DEMO
- DataWedge profile is configured to send intents
- Take action when an Intent (barcode) is received:

```
EB.Intent.startListening(
function(intent) {barcodeAsJson = intent.data;
alert(barcodeAsJson['com.symbol.datawedge.data_string']);});
```

- Enable or disable the scanner with the DW intent API depending on the page shown:

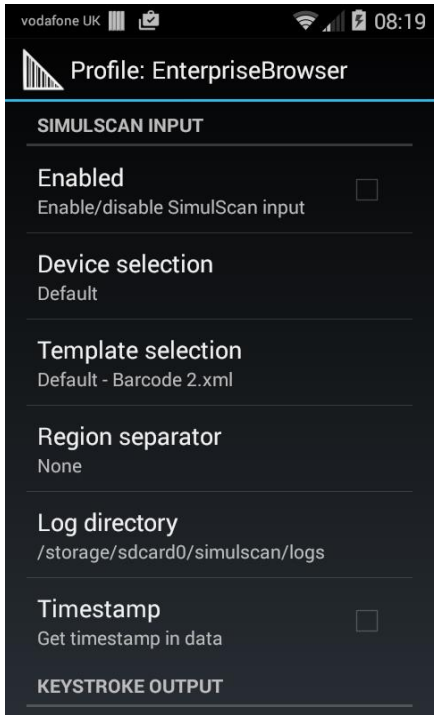
```
EB.Intent.send(
{'intentType':EB.Intent.BROADCAST,
'action':'com.symbol.datawedge.api.ACTION_SCANNERINPUTPLUGIN',
'data':{'com.symbol.datawedge.api.EXTRA_PARAMETER':'ENABLE_PLUGIN'}});
```

ZEBRA

Modifying the Application

Principle: SimulScan

- SimulScan is part of Zebra's Mobility DNA and enables simultaneous capture of barcodes and images as well as optical character and mark recognition.
- More powerful functionality such as OCR / OMR requires a license but barcode capture can be used FOC
- SimulScan offers a DataWedge interface and an EMDK API (but no API in EB yet). We will be using the DataWedge interface
- You can define your own templates but several come pre-installed including a "2 barcode" template which we can use.



ZEBRA

Modifying the Application

Demo: SimulScan

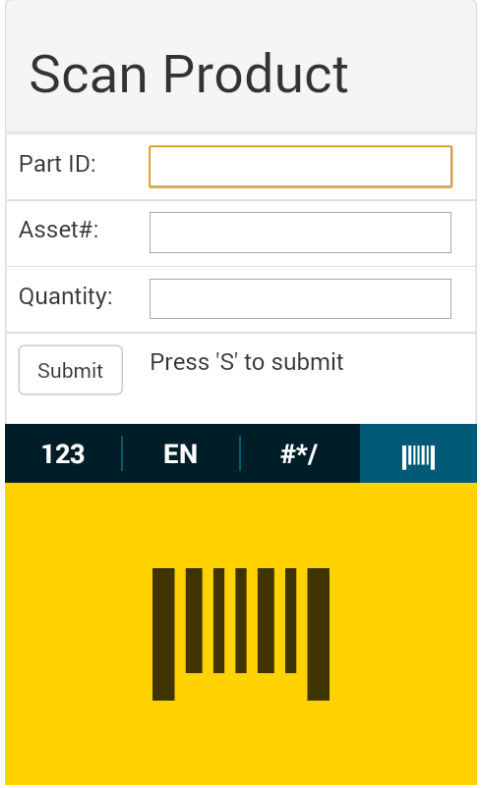
- DEMO
- Additional code in our application to check for a ‘SimulScan’ data capture:

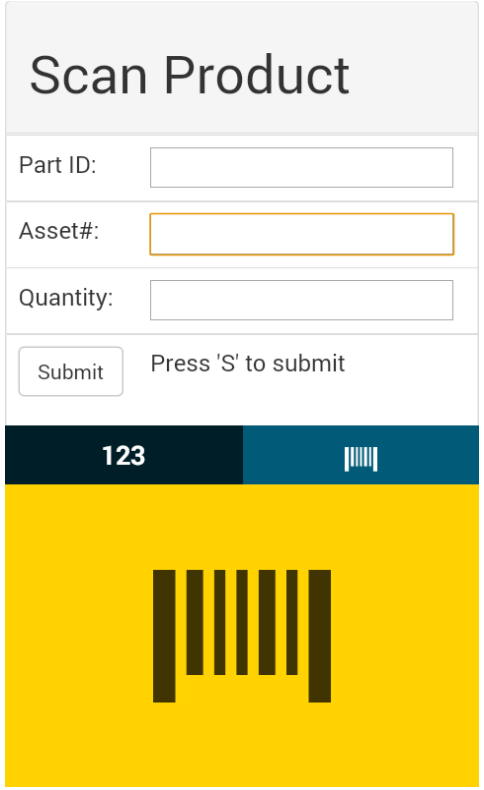
```
if (source == 'simulscan'){
    //  Handle SimulScan Data
    decodedRegions = barcodeJson['com.symbol.datawedge.simulscan_region_data'];
    barcode1 = decodedRegions[0];
    barcode1Data = barcode1['com.symbol.datawedge.simulscan_region_string_data'];
    barcode2 = decodedRegions[1];
    barcode2Data = barcode2['com.symbol.datawedge.simulscan_region_string_data'];
    receivedBarcode("" + barcode1Data);
    receivedBarcode("" + barcode2Data);}
```

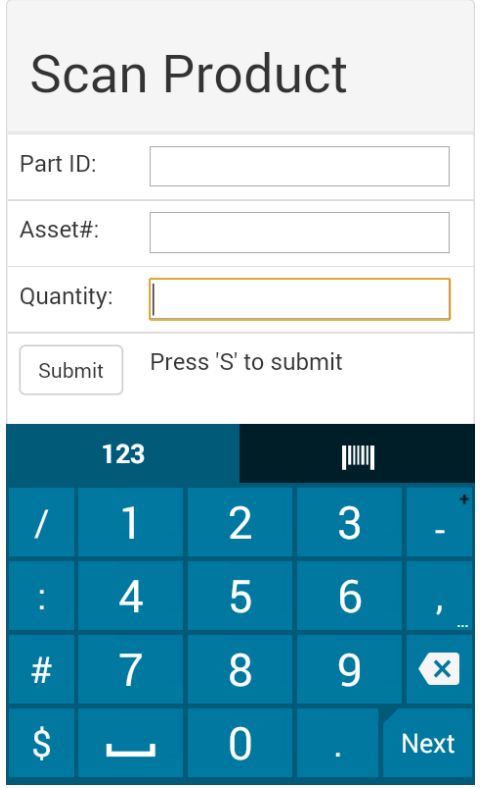
Modifying the Application

Principle: Enterprise Keyboard Integration

- Goal is to change the keyboard layout depending on the format of the text field
- Technology preview, this will be making its way into the product soon







Modifying the Application

Demo: Enterprise Keyboard Integration

- DEMO
- Engineering version of EKB installed for this demo
- DOM injection required to set “type” of input fields: text or number. “Z_method” can be used to denote a scan field
- Add eventlistener() to all input elements:
  - Show SIP through Enterprise Browser API: EB.Sip.show();
  - Open custom URI zebra://input?type=blah. This will be received by the Enterprise Keyboard and the layout changed accordingly.


ZEBRA
APPFORUM 2017
INNOVATE. ENGAGE. TRANSFORM.
Questions?

Please take a moment to rate this session using the APPFORUM mobile app.

THANK YOU

Content Slide Title Goes Here

Sub title goes here

- Bullet text