Zebra DevCon 2023

ZEBRA

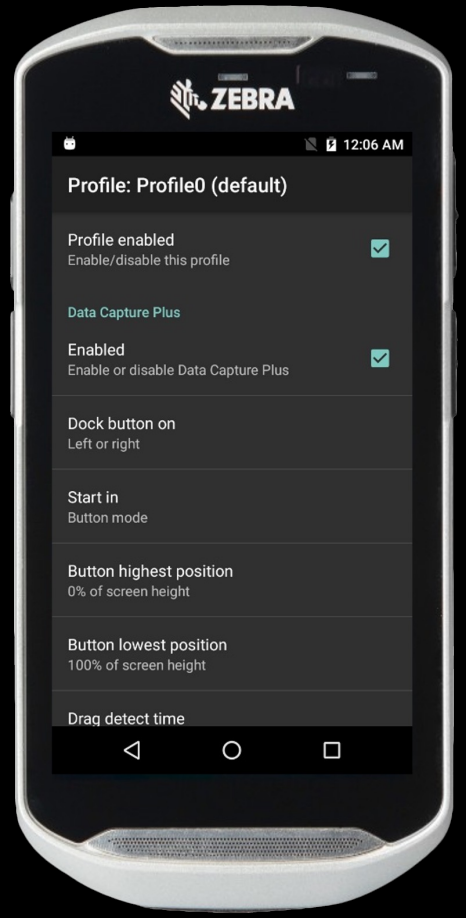
Capture More with DataWedge

Prashanth Kadur

DataWedge

Zebra DevCon 2023

Zebra's DataWedge provides the capability for any application on the device to easily acquire data (barcode(s), images, OCR and more) from various input sources (such as barcode scanner, RFID, voice, and serial port) and manipulate the data as needed based on simple or complex rules.



Application / Activity based DataWedge profiles can be configured to controls:

- To enable/disable data capturing
- Select data capture source and their configurations
- How data needs to be processed
- How data needs to be dispatched to the application(s)

Profiles can be deployed via MDM or staging processes

Developers can write their applications to configure and control DataWedge to full fill customer requirements.

What's new in DataWedge

Zebra DevCon 2023

OCR Wedge	Capture VIN, TIN, Container ID, Identification Documents, Meter, License Plate Reading using the Rear Camera
Barcode Highlighting	Highlight and Capture barcode(s) in the field of view based on defined rules using the Integrated Imager or Rear Camera
Free-Form Image Capture	Capture images using the Integrated Imager or Rear Camera
NG SimulScan template integration	NG SimulScan templates can be integrated via Intent APIs
Free-Form OCR	Capture OCR within the on-screen resizable viewfinder powered by Google ML KIT Text Recognition v2 (Beta) API

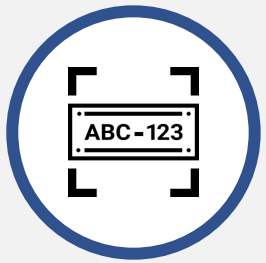
What's new in DataWedge

Zebra DevCon 2023

QRCode Launching	QRCode web links can be auto launched
Beep notification in ADF actions	Audio feedback for user when specific barcodes are scanned
Use Two Barcode scanners	Simultaneously use built in scanner and an external scanner
Access Control to DataWedge Intent APIs	Restrict DataWedge Intent API usage for specific applications.
Using DataWedge in WS50	How to use DataWedge in Zebra's latest wearable device

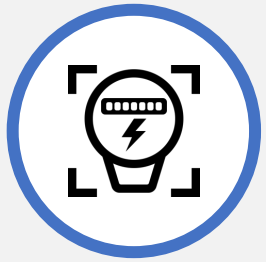
The Solution

Six powerful OCR Wedge configurations automatically recognize and capture physical text, transforming it into accurate digital data that streams right into your application



License Plates

Container Identification Numbers



Meter Reading

Tire Identification Numbers (TIN)

Identification Documents

Vehicle Identification Numbers (VIN)

Each Configuration Available in 1-Year and 2-Year Term Licenses

Features

OCR AND MACHINE LEARNING

Extracts only the text you need, Fine-tuned and trained to overcome specific challenges, from text that is curved or low contrast to ignoring surrounding text

100% OFF-LINE ON DEVICE PROCESSING

No Wi-Fi or cellular connection required - Works anywhere – including remote locations and underground parking garages

AUTOMATIC POINT-AND-SHOOT SIMPLICITY

Rapid character recognition, just point and center the camera on the desired text – OCR Wedge does the rest

DEPLOY QUICKLY AND EASILY

Powered by DataWedge, just choose your OCR features, purchase the license and get up and running in minutes

Benefits

Increased productivity

Automation eliminates steps, allowing workers to complete more tasks and more work orders per day

Increased task accuracy

Eliminate manual data entry errors, improving customer service and customer safety

A superior user experience

Data capture is virtually effortless — no more tedious and time-consuming manual data entry

Instant data access

Faster access to information enables better decisions and faster actions

The Solution

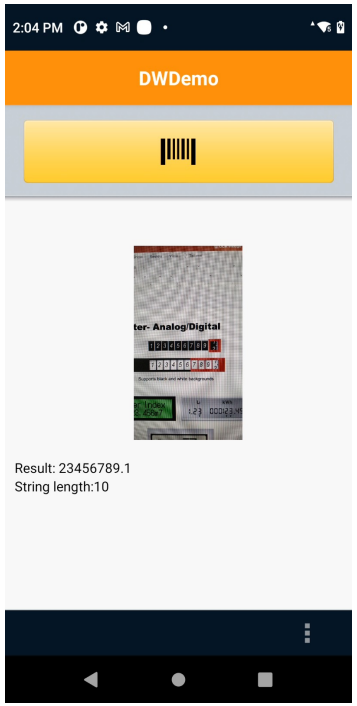
Zebra Mobility DNA OCR Wedge

Zebra DevCon 2023

LICENSE PLATES

CONTAINER IDENTIFICATION NUMBERS

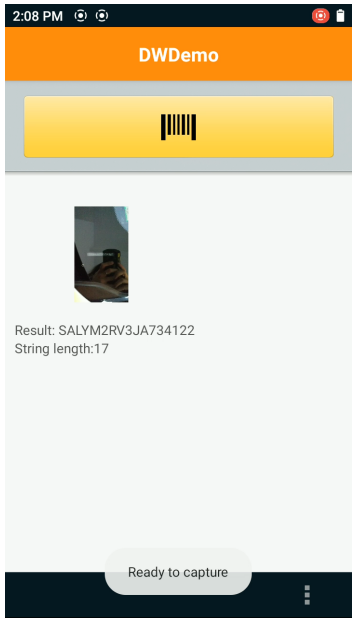
METER READING



TIRE IDENTIFICATION NUMBERS (TIN)

IDENTIFICATION DOCUMENTS

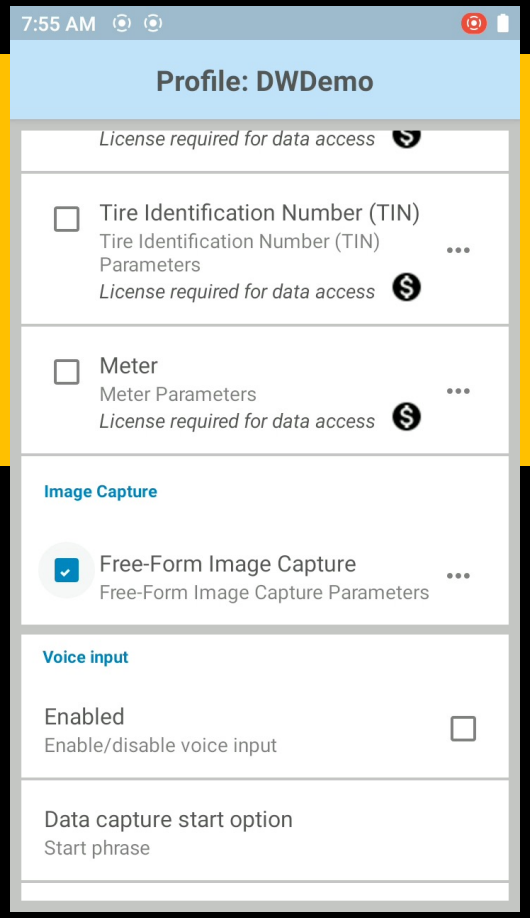
VEHICLE IDENTIFICATION NUMBERS (VIN)



Free-Form Image Capture

No camera? No Problem – Use the integrated imager (or rear camera) to quickly capture an image

Zebra DevCon 2023

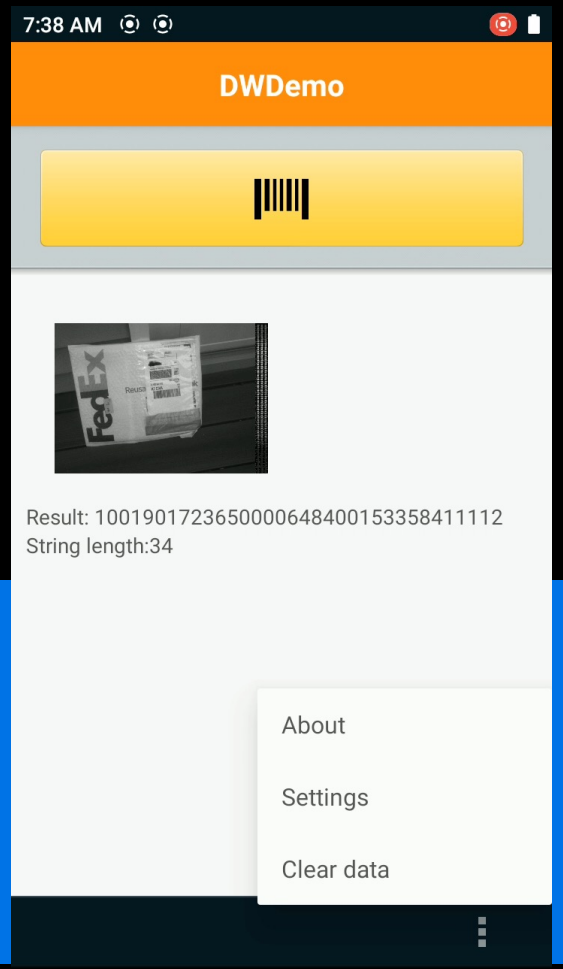


How to Configure?

1. Ensure Workflow Input is enabled
2. Check off “Free-Form Image Capture”
3. Set Parameters (optional)
   - Decode & Highlight Barcodes (Option for Off, Highlight Only, or Highlight and Decode)
   - Illumination (option for on/off) for near-range/ darker environments- illumination may be optimal

User Interaction:

1. Soft Button or Hard trigger initiates Session
2. As system sees barcodes it will highlight them to let user know it has been seen
3. The next trigger will capture the image, so the user can step back, if need be, to ensure entire object is within view
4. If Decode/highlight option is turned on it will return the image + the barcodes it had highlighted



▲ **Video How to configure**

▲ **Video See it in action!**

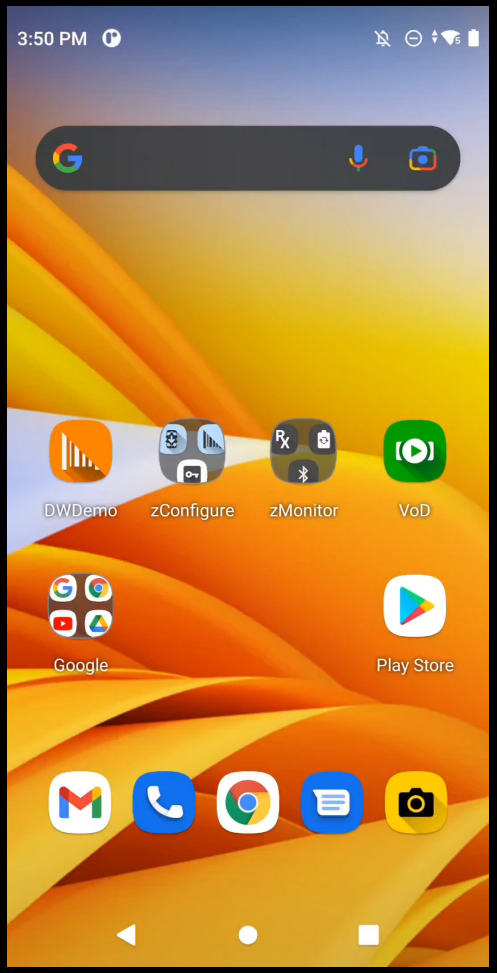
Free-Form OCR

Zebra DevCon 2023

Able to capture text using camera or integrated scan engine.

Supported via “Google ML Tool Kit V2” text recognition feature.

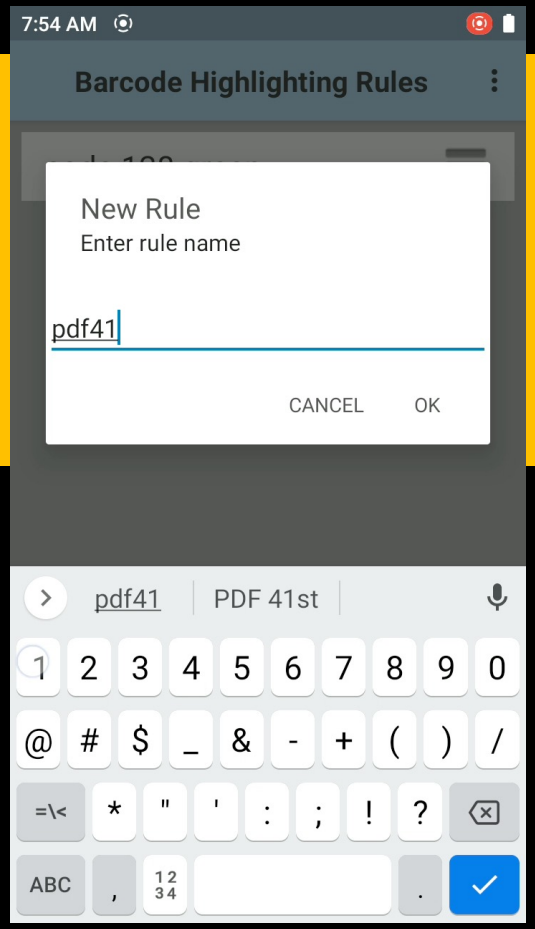
Free-Form OCR is powered by Google ML Kit Text Recognition v2 (Beta) API. Per Google, "Text Recognition v2 is offered in beta, which means it might be changed in backward-incompatible ways and is not subject to any SLA or deprecation policy." (source)



Barcode Highlighting

Highlight items to let the user know when they have found what they need

Zebra DevCon 2023



How to Configure?

1. Ensure Barcode Input is enabled
   - Scanner Selection **can be configured for Camera or Imager!!**
2. Check off “Barcode Highlighting”
3. Set Barcode Highlighting Parameters
   - Barcode Highlighting Rules (this tells what barcodes to highlight)
   - Report Data Rules (this tells which barcodes to report back)

User Interaction:

1. Soft Button or Hard trigger initiates Session
2. As system sees barcodes it will highlight them as defined in the barcode highlighting rules
3. The next trigger will end the session and return the decode data of the barcodes defined in the report data rules

Video How to configure

Video See it in action!

Barcode Highlighting

Programmatically configure the rules based on application requirements

Zebra DevCon 2023

```
public void highlightBarcodes()
{
    Intent i = new Intent();
    i.setAction("com.symbol.datawedge.api.ACTION");
    i.putExtra("APPLICATION_PACKAGE", getPackageName());
    i.setPackage("com.symbol.datawedge");
    i.putExtra("SEND_RESULT","LAST_RESULT");
    i.putExtra("com.symbol.datawedge.api.SWITCH_DATACAPTURE", "BARCODE");

    Bundle paramList = new Bundle();
    paramList.putString("scanner_selection_by_identifier","INTERNAL_IMAGER");
    paramList.putString("barcode_tracking_enabled","true");

    Bundle rule1 = new Bundle();
    rule1.putString("rule_name","Rule1");
    Bundle rule1Criteria = new Bundle();

    rule1Criteria.putStringArray("symbology",new String[]{"decoder_code128"});
    rule1.putBundle("criteria",rule1Criteria);

    Bundle bundleFillColor = new Bundle();
    bundleFillColor.putString("action_key","fillcolor");
    bundleFillColor.putString("action_value","#78FF0F00");

    ArrayList<Bundle> rule1Actions = new ArrayList<>();
    rule1Actions.add(bundleFillColor);

    rule1.putParcelableArrayList("actions",rule1Actions);

   //Continue....
```

```
  //Continue from previous section....

    Bundle rule2 = new Bundle();
    rule2.putString("rule_name","Rule2");
    Bundle rule2Criteria = new Bundle();
    rule2Criteria.putStringArray("symbology",new String[]{"decoder_code39"});
    rule2.putBundle("criteria",rule2Criteria);
    Bundle rule2BundleStrokeColor = new Bundle();
    rule2BundleStrokeColor.putString("action_key","fillcolor");
    rule2BundleStrokeColor.putString("action_value","#780002FF");
    ArrayList<Bundle> rule2Actions = new ArrayList<>();
    rule2Actions.add(rule2BundleStrokeColor);
    rule2.putParcelableArrayList("actions",rule2Actions);


    ArrayList<Bundle> ruleList = new ArrayList<>();
    ruleList.add(rule1);
    ruleList.add(rule2);

    Bundle ruleBundlebarcodeOverlay = new Bundle();
    ruleBundlebarcodeOverlay.putString("rule_param_id","barcode_overlay");
    ruleBundlebarcodeOverlay.putParcelableArrayList("rule_list", ruleList);

    ArrayList<Bundle> ruleParamList = new ArrayList<>();
    ruleParamList.add(ruleBundlebarcodeOverlay);

    paramList.putParcelableArrayList("rules", ruleParamList);

    i.putExtra("PARAM_LIST", paramList);

    sendBroadcast(i);
}
```

Use DataWedge WS50

Zebra DevCon 2023

DataWedge on WS50 has no UI for editing a profile

Deploy Configuration

- Create the profile(s) on another Zebra device such as any TC5x
- Export the profile into a .db file via the Datawedge Settings menu
- Copy this file to a PC and then push it to the WS50 device via ADB
- *adb push dwprofile_myprofilename.db /enterprise/device/settings/datawedge/autoimport*

Profile can also be copied to autoimport folder via Mx FileMgr

Applications can use DataWedge Intent APIs to configure and control DataWegde.

Recommended to use Intent Output to receive data to applications.

https://techdocs.zebra.com/datawedge/11-3/guide/programmers-guides/ws50/

NG Simulscan – Multi-Barcode

Powerful MultiBarcode Features to automate data capture that streams right into your application

ON-DEVICE CONFIGURATIONS

ANY BARCODE: No rules to define specific barcodes to capture.

Fixed Quantity
Capture a defined number of barcodes from session to session.

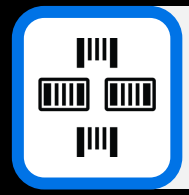
Variable Quantity*
Capture a changing number of barcodes from session to session.

On-Device MultiBarcode Parameters:

- **Instant Reporting:** Parameter allows user to control multibarcode scan sessions. Decoded feedback flows back instantly for every single barcode as an individual string.
- **Report Decoded Barcodes:** Parameter is what changes Fixed Quantity into Variable Quantity.

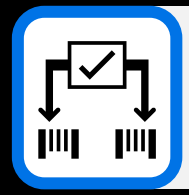
NG SIMULSCAN TEMPLATE BUILDER CONFIGURATIONS

SPECIFIC BARCODE(S): *Unique criteria distinguishes specific barcodes from other barcodes within a scan session.*



Specific Barcodes*
Capture specific barcodes based on unique criteria (symbology, starting character(s), length etc.).

GROUP OF COMMON BARCODES:
More than one Barcode that shares a common pattern



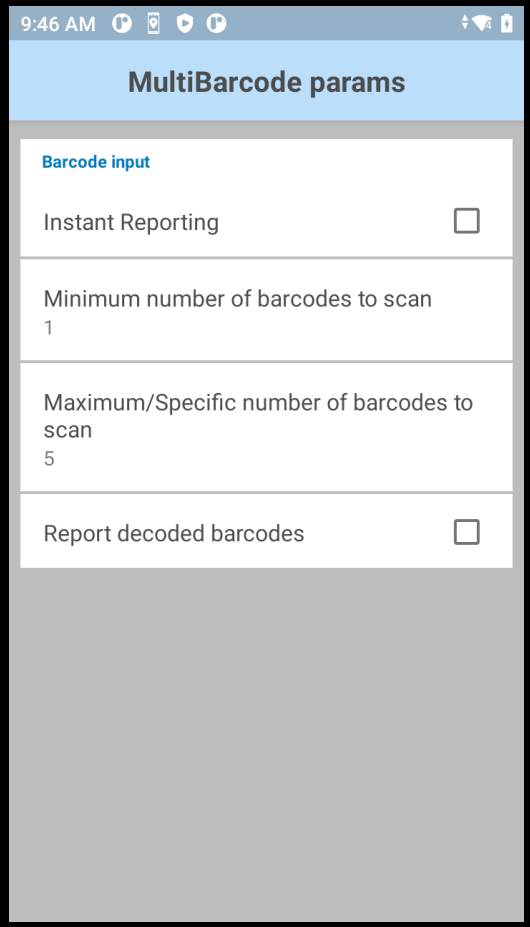
Automatic Group Identification*
Capture changing number of barcodes that all share the same unique criteria (symbology, starting character(s), length etc.).

*Only supported in Internal image or camera scanner

Multi-Barcode

Zebra DevCon 2023

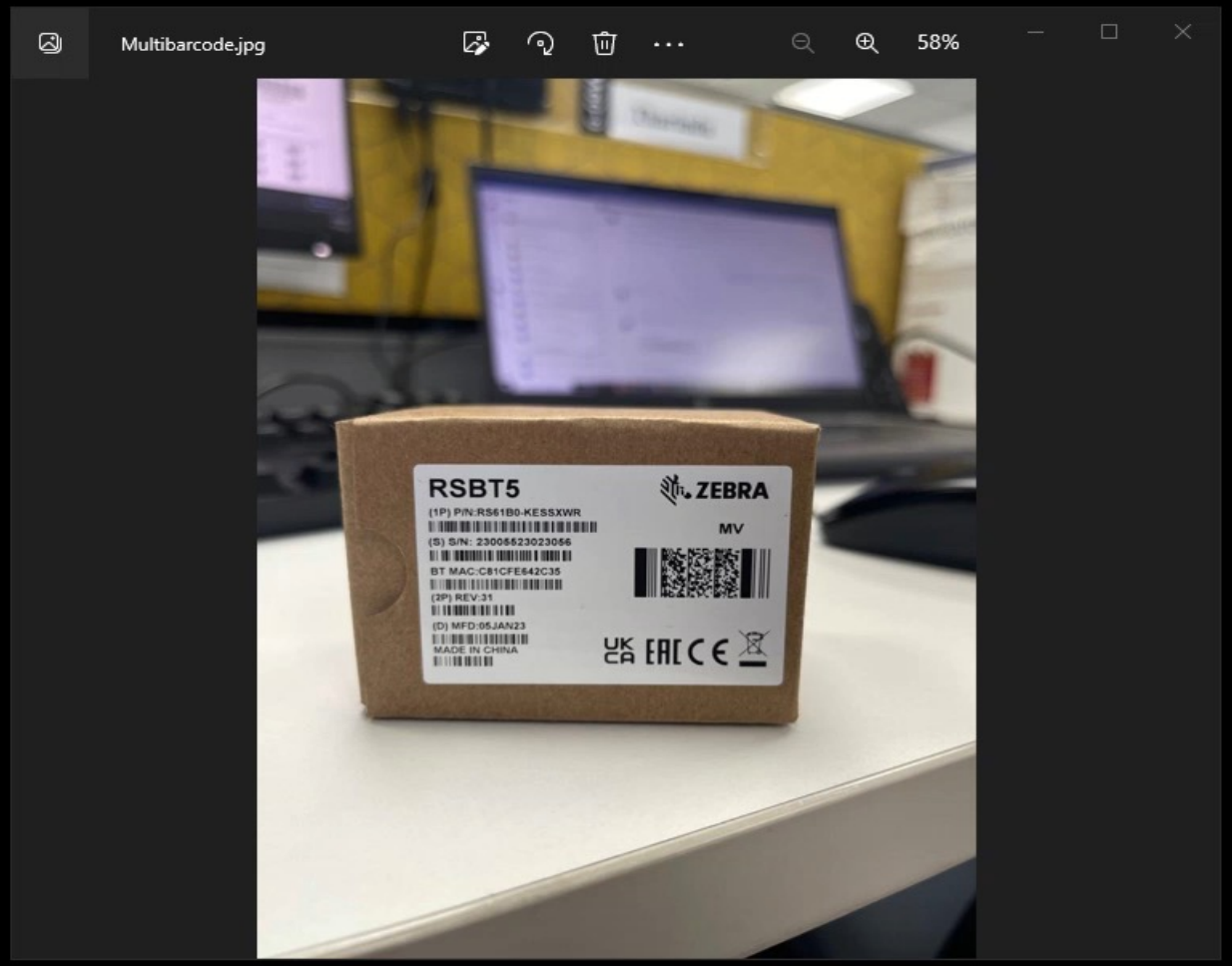
- Applications can scan multiple barcodes in a single scanning session by pressing and holding the trigger button
- When multi-barcode option is enabled, user can scan up to 100 barcodes (without duplicates) depending on the set configuration
- Configurable options
  - Instant Reporting - Enable/Disable immediate reporting of unique barcodes within a scan session
  - Minimum barcode count* - Supported values 1 to 100
  - Maximum barcode count - Supported values 1 to 100
  - Report decoded barcodes* - Allows reporting of decoded barcodes in a single scan session irrespective of the specified Maximum/Specific number of barcodes per scan
- It is recommended to use Intent output to receive Multi-barcode data as developers have access to meta data such as (Label type) where applications can take advanced decisions
- If key stroke out put is used profile can be configured to add barcode separator to separate the barcodes with a Line Feed, Carriage Return or a Tab character



** Only supported in Internal image or camera scanner*

Multi-Barcode (Demo)

Zebra DevCon 2023



Multi-Barcode

Zebra DevCon 2023

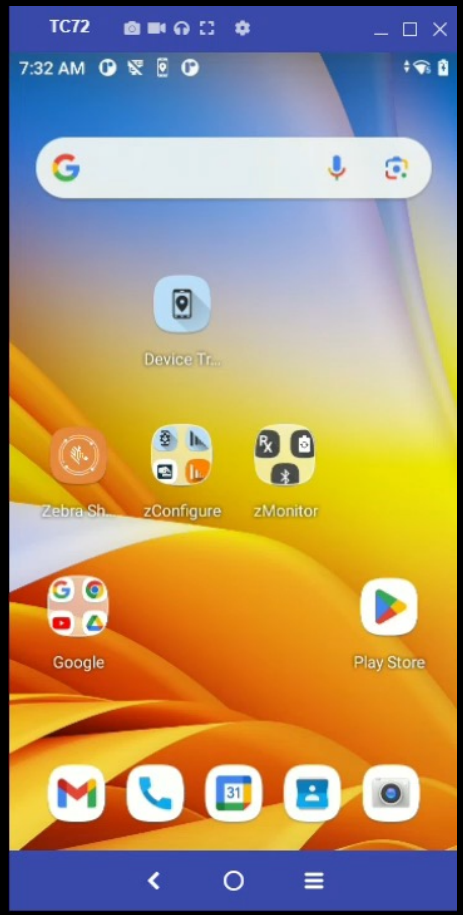
```
private BroadcastReceiver myBroadcastReceiver = new BroadcastReceiver() {
   @Override
   public void onReceive(Context context, Intent intent) {
      String action = intent.getAction();
      Bundle b = intent.getExtras();

      if (action.equals(PROFILE_INTENT_ACTION))
      {
         //  Received a barcode scan
         displayScanResult(intent);
      }
   }
};
```

```
private void displayScanResult(Intent intent)
{
   TextView txtOutput = findViewById(R.id.txtOutput);
   String output = txtOutput.getText().toString();
   String decoded_mode =     intent.getStringExtra("com.symbol.datawedge.decoded_mode");
   if (decoded_mode.equalsIgnoreCase("multiple_decode"))
   {
      String barcodeBlock = "";
      List<Bundle> multiple_barcodes = (List<Bundle>)
                     intent.getSerializableExtra("com.symbol.datawedge.barcodes");
      if (multiple_barcodes != null)
      {
         for (int i = 0; i < multiple_barcodes.size(); i++)
         {
            Bundle thisBarcode = multiple_barcodes.get(i);
            String barcodeData = thisBarcode.getString("com.symbol.datawedge.data_string");
            String symbology = thisBarcode.getString("com.symbol.datawedge.label_type");
            barcodeBlock += "Barcode: " + barcodeData + " [" + symbology + "]";
            if (multiple_barcodes.size() != 1)
               barcodeBlock += "\n";
         }
      }
      txtOutput.setText(barcodeBlock + "\n" + output);
   }
}
```

UDI

Zebra DevCon 2023

- UDI (Unique Device Identifier) : FDA standard used in Healthcare Industry
- UDI barcodes are created with information about an item. They contain Device Identifier (**DI**) and Production identifiers(**PIs**)
  - **DI** - a mandatory, fixed portion of a UDI that identifies the labeler and the specific version or model of a device
  - **PI** - a conditional, variable portion of a UDI that identifies one or more of the following
    - Lot or batch number
    - Serial number
    - Expiration date
    - Date of manufacturing
- DataWedge supports UDI barcode capturing for the following standards
  - ***GS1***
  - ***HIBCC***
  - ***ICCBBA***
- UDI data can be received as keystroke data or Intent data. Intent data will have more detailed data
- Beyond UDI – Regular GS1 Parsing (New Feature Available from October 2023)
  - Non-UDI standard GS1 barcodes will be able to capture and receive them as tokens like UDI data



UDI

Receiving UDI data to the application

Zebra DevCon 2023

```
private BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
   @Override
   public void onReceive(Context context, Intent intent) {

      if (intent.getAction().equalsIgnoreCase(INTENT_OUTPUT_ACTION)){
         Bundle bundle = intent.getExtras();
         StringBuilder sb = new StringBuilder();
         ArrayList<Bundle> arrayList = bundle.getParcelableArrayList("com.symbol.datawedge.tokenized_data");
         for(Bundle data: arrayList){
            String result = data.getString("token_string_data");
            String tokenId = data.getString("token_id");
            sb.append(tokenId)
                  .append("\t \t")
                  .append("-")
                  .append("\t \t").append(result);
            sb.append("\n");
         }
         txt.setText(sb);
      }
   }
};
```

DataWedge Intent APIs

Zebra DevCon 2023

1. DataWedge Intent APIs have been around for several years
2. The Intent APIs were introduced to allow developers to programmatically configure, control DataWedge and receive important application-level notifications from DataWedge
3. Advantage of DataWedge intent APIs is that
    1. No special dependencies need to be added to the development environment
    2. Administrator has the control to allow only the set of whitelisted applications can make use of DataWedge Intent APIs
    3. All data capture functionality is supported
4. Zebra strongly encourage developers to use DataWedge intent APIs over EMDK APIs as they support the full set of Data Capture functionality

DataWedge Intent APIs

Zebra DevCon 2023

Configuration APIs

- **Clone Profile** - Create a copy of an existing DataWedge Profile including all settings.
- **Create Profile** - Create a new Profile without setting configurations.
- **Delete Profile** - Delete an existing Profile.
- **Import Config** - Import a Profile and/or Config file, which can contain multiple Profiles.
- **Rename Profile** - Rename an existing Profile.
- **Restore Config** - Reset all user-configured settings and restore DataWedge to the factory-default settings.
- **Set Config** - Create, update or replace a Profile and its settings.
- **Set Disabled App List** - Add, remove or update an app/activity from the list which prevents the use of DataWedge.
- **Set Ignore Disabled Profiles** - Prevent switching to a Profile that is disabled, including Profile0.

Controlling APIs

- **Enable/Disable DataWedge** - Enable/disable DataWedge on the device.
- **Enable/Disable Scanner Input Plug-in** - Enable/Disable the Scanner Input Plug-in in use by the current active profile, which effectively disables scanning.
- **Enumerate Triggers** - Retrieve supported trigger list of a device.
- **Enumerate Workflows** - Retrieve supported workflows on a device.
- **Notify (Bluetooth Scanner Notifications)** - Play notification sound(s) and/or display a colored LED after a scan from a connected Bluetooth scanner with RSM (Remote Scanner Management).
- **Reset Default Profile** - Reset the default profile to Profile0, the built-in profile used with unassociated apps.
- **Set Default Profile** - Set the specified profile as the default profile.
- **Soft RFID Trigger** - Start, stop or toggle a software RFID trigger.
- **Soft Scan Trigger** - Start, stop or toggle a software scanning trigger.
- **Soft Trigger** - Start, stop or toggle a software trigger for voice input.
- **Switch Data Capture** - Switch between barcode scanning and barcode highlighting, or switch between Workflow Input options at runtime.
- **Switch Scanner** - Switch to a specific scanner at runtime to enable an optimal scanning device for the app, requirement or situation.
- **Switch Scanner Params** - Temporarily update the settings of the active profile during runtime by passing one or more barcode, scanner and/or reader parameters as intent extras.
- **Switch to Profile** - Change the app association to the specified profile.

DataWedge Intent APIs

Zebra DevCon 2023

Query APIs

- **Enumerate Scanners** - Generate an index of scanners available on the device.
- **Get Active Profile** - Get the name of the profile currently in use by DataWedge.
- **Get Config** - Gets the `PARAM_LIST` settings in the specified Profile, returned as a set of value pairs or a Plug-in config bundle.
- **Get DataWedge Status** - Return the DataWedge status, enabled or disabled.
- **Get Disabled App List** - Return a list of apps and activities that are blocked from using DataWedge.
- **Get Ignore Disabled Profiles** - Return the status of the "Ignore Disabled Profiles" parameter. If true, DataWedge cannot switch to any profile that is not enabled.
- **Get Profiles List** - Return the list of DataWedge profiles.
- **Get Scanner Status** - Return the status of the scanner currently selected as the default.
- **Get Version Info** - Return the current version of DataWedge and Scanner Framework/Decoder library installed on the device.

Notification APIs

- **Register/Unregister for Notifications** enables apps to register or unregister to receive notifications of status changes related to:
  - Profile switching
  - Scanner status
  - Configuration
  - Workflow status - also applies to Barcode Highlighting

DataWedge Intent APIs

Zebra DevCon 2023

```
    // MAIN BUNDLE PROPERTIES
    Bundle bMain = new Bundle();
    bMain.putString("PROFILE_NAME","Profile12");          // <- "Profile12" is a bundle
    bMain.putString("PROFILE_ENABLED","true");            // <- that will be enabled
    bMain.putString("CONFIG_MODE","CREATE_IF_NOT_EXIST");  // <- or created if necessary.

// PLUGIN_CONFIG BUNDLE PROPERTIES
    Bundle bConfig = new Bundle();
    bConfig.putString("PLUGIN_NAME","BARCODE");
    bConfig.putString("RESET_CONFIG","true");


// PARAM_LIST BUNDLE PROPERTIES
    Bundle bParams = new Bundle();
    bParams.putString("scanner_selection","auto");
    bParams.putString("scanner_input_enabled","true");
//
// NOTE: The "scanner_selection" parameter (above) supports "auto" selection
// --OR-- the assignment of a scanner device index, which is obtained by
// using the ENUMERATE_SCANNERS API.
//
//      Syntax for scanner index:
//
//          Bundle bParams = new Bundle();
//      diff--> bParams.putString("current-device-id","0");
//          bParams.putString("scanner_input_enabled","true");
//
//
// NEST THE BUNDLE "bParams" WITHIN THE BUNDLE "bConfig"
    bConfig.putBundle("PARAM_LIST", bParams);

// THEN NEST THE "bConfig" BUNDLE WITHIN THE MAIN BUNDLE "bMain"
    bMain.putBundle("PLUGIN_CONFIG", bConfig);
```

```
// CREATE APP_LIST BUNDLES (apps and/or activities to be associated with the Profile)
    Bundle bundleApp1 = new Bundle();
    bundleApp1.putString("PACKAGE_NAME","com.symbol.emdk.simulscansample1");
    bundleApp1.putStringArray("ACTIVITY_LIST", new String[]{
        "com.symbol.emdk.simulscansample1.DeviceControl",
        "com.symbol.emdk.simulscansample1.MainActivity",
        "com.symbol.emdk.simulscansample1.ResultsActivity.*",
        "com.symbol.emdk.simulscansample1.ResultsActivity2",
        "com.symbol.emdk.simulscansample1.SettingsFragment1"});

    Bundle bundleApp2 = new Bundle();
    bundleApp2.putString("PACKAGE_NAME","com.example.intents.datawedgeintent");
    bundleApp2.putStringArray("ACTIVITY_LIST", new String[]{
        "com.example.intents.datawedgeintent.DeviceControl",
        "com.example.intents.datawedgeintent.MainActivity",
        "com.example.intents.datawedgeintent.ResultsActivity",
        "com.example.intents.datawedgeintent.SettingsFragment1"});

    Bundle bundleApp3 = new Bundle();
    bundleApp3.putString("PACKAGE_NAME","com.symbol.myzebraapp");
    bundleApp3.putStringArray("ACTIVITY_LIST", new String[]{"*"});

// NEXT APP_LIST BUNDLE(S) INTO THE MAIN BUNDLE
    bMain.putParcelableArray("APP_LIST", new Bundle[]{
        bundleApp1
        ,bundleApp2
        ,bundleApp3
    });

    Intent i = new Intent();
    i.setAction("com.symbol.datawedge.api.ACTION");
    i.putExtra("com.symbol.datawedge.api.SET_CONFIG", bMain);

    this.sendBroadcast(i);
```

Securing DataWedge Intents

Zebra DevCon 2023

DataWedge provides an option for admins to control access to DataWedge intent APIs, allowing only approved apps to configure DataWedge.

DataWedge intent APIs are categorized, allowing the administrator to grant DataWedge API access to specific apps based on category. By default, DataWedge accepts any intent API to avoid impact to existing applications.

Refer to https://techdocs.zebra.com/datawedge/11-3/guide/programmers-guides/secure-intent-apis/

Configuration Deployment

Zebra DevCon 2023

1. DataWedge configuration can be exported from a configured device and mass deployed to other devices
2. Configuration files come in the form of .db files which can have either a full configuration of DataWedge or an individual profile.
3. Configuration Deployment can be
   - Manual
   - Automated
4. Manual options (For Developers)
   - Manually export and import the configuration files via configuration UI
   - Copy the configuration files to /enterprise/device/settings/autoimport folder. (This is not recommended due to security concerns)
5. Automated options (For production)
   - Use EMMs to deploy exported configuration files via Zebra's File Manager
   - Use EMMs to deploy via DataWedge Manager

Configuration Deployment

Via File Manager

Customize Soft Keyboard with Zebra EKB

Zebra DevCon 2023

- Enterprise KeyBoard (EKB) is Zebra IME which provides the user the ability to create a custom soft keyboard layout via Enterprise Keyboard Designer
- If EKB is installed and set as the default IME of a Zebra Android mobile computer developers can configure different keyboard layouts to DataWedge profiles
- Based on the currently running application/activity, the configured keyboard layout will be presented to the user for their data input

Showcase Case Demo

16 Data Capture “Demos”

Zebra DevCon 2023

SCANNING

(3)

Any Barcode
Scan 1D and 2D Barcodes

Picklist Mode
Selective Barcode Capture

UDI Healthcare Barcodes
Capture & Parse UDI Barcodes

New Platform for showcasing our offerings- from features to solutions. Allows Zebra Device holders to learn about our solution with preconfigured demos. Built on EB, no licenses required, to offer 16 Data Capture Demos at Launch. Launch will also come with the developer.

MULTI-BARCODE

(3)

Fixed or Variable Quantity
Capture a changing number of barcodes

Specific Barcodes
Capture specific barcodes from any Zebra Label

Automatic Group ID
Automatically identifies & capture a group of common barcodes like serial numbers

IMAGE

(2)

Free-Form Image Capture
Quickly capture an image even with the integrated Imager

Whole Document Capture
Capture & crop document with optional barcode

OCR

(8)

Free-Form OCR
Extract printed text from labels, documents & more

TIN OCR Wedge
Capture Tire ID numbers molded into tire sidewalls

License Plate OCR Wedge
Capture US & EU Vehicle License Plates

VIN OCR Wedge
Capture Vehicle ID numbers from paper & windshields

ID OCR Wedge
Capture drivers' licenses & identification cards

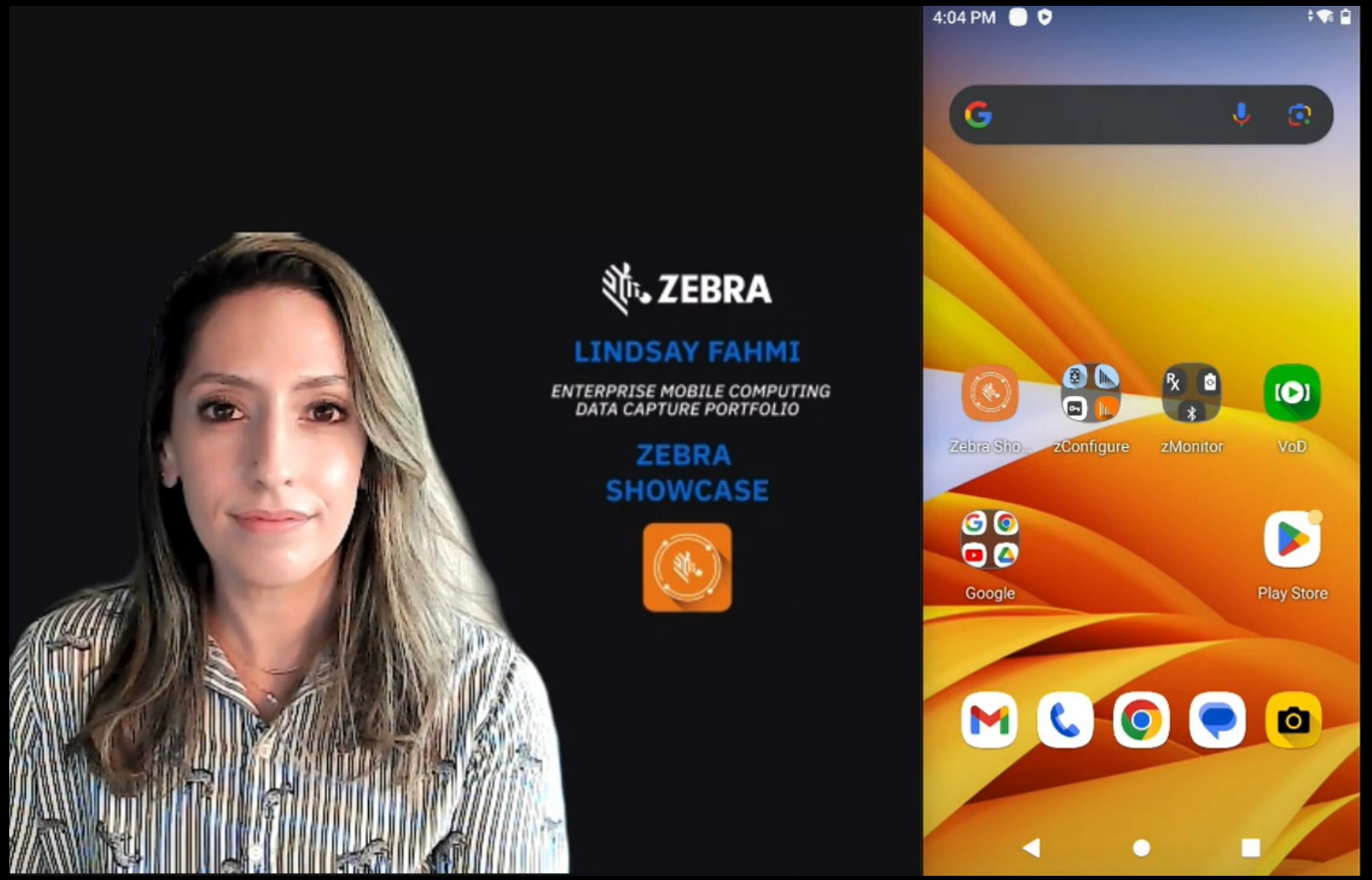
OCR-B Travel Documents
Scan MRZ of passports and Visas

Meter Reading OCR Wedge
Automatically read analog, digital, and dial meters

Container OCR Wedge
Capture Id Numbers from shipping containers

Showcase Demo App

Zebra DevCon 2023



What's Store for the Future

Zebra DevCon 2023

- DataWedge in Cope Mode
- MP7000 Scale Support
- Zebra's ZAVS (Zebra Adopted Vision Software)

DataWedge COPE Mode

Zebra DevCon 2023

- Zebra Android Mobile computers going to support COPE mode (**C**operate **O**wned **P**ersonally **E**nabled)
- Customers can provide a Zebra Mobile Computer to their associates to do the **enterprise work** as well as use it for their **personal use**
- There will be two user profiles in the device
  - Personal – Associate can use for their personal use
  - Work – Administrator will install all the enterprise applications

DataWedge COPE Mode

Zebra DevCon 2023

DataWedge will be supported in the COPE mode but there are few things need to know

- By default, DataWedge won't be enabled in the Work Profile
- Administrator must enable the DataWedge application in the Work profile
- When DataWedge is enabled in the Work Profile user will not be able to use DataWedge in personal user apps

Supported Features

- Barcode scanning via internal imager
- Keystroke data dispatching
- Intent data dispatching
- Dispatching data using IP Output
- Data Capture Plus
- Advanced data formatting
- Basic data formatting
- Import/Export DataWedge configurations Mass deployment

Limitations

- Cannot use SD card for configuration deployment
- Un supported features
  - Voice Input
  - Serial Input
  - Enable scanning in Launcher screen

MP7000 Scale Support

Zebra DevCon 2023

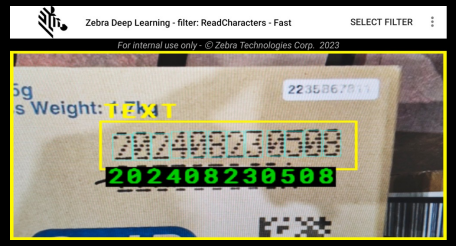
- MP7000 is a Grocery Scanner Scale
- Will be ideal for point of Sales counter with Zebra Workstation Connect
- MP7000 can be used for barcode scanning as well as measuring the weight of items at the checkout counter
- DataWedge going add support for applications to read the weight of an object once placed on the MP7000

Zebra's ZAVS (Zebra Adopted Vision Software)

Interest in ZAVS Integration in Zebra Mobile Computers? Let us know!

Zebra DevCon 2023

- **OCR**
  - Optical Mark Recognition - Logo detection
- **ADVANCED OCR**
  - Advanced Fonts like Dot Peen



- **QUALITY INSPECTION**
  - Leaflet inspection
  - Packaging
  - Label Inspection
  - Box Completeness
- **PRODUCT DETECTION**
  - Product Classification
  - Counting

Questions?

Zebra DevCon 2023


https://developer.zebra.com


Zebra Developers Community – LinkedIn Group


@ZebraDevs


https://github.com/ZebraDevs

Zebra DevCon 2023

ZEBRA

Thank You