DARRYN CAMPBELL

Mobile computing and enterprise software development

Tutorial: The DataWedge Intent API

3rd January 2018 14 By DARRYNCAMPBELL

This tutorial will show the basic principles of how to control DataWedge via the DataWedge Intent API. The DataWedge API can be quite powerful, I have written a fully featured application in the past but frequently I find people looking for something simpler as a jumping off point.

This tutorial requires a Zebra device running DataWedge 6.3 or higher.

DataWedge configuration

There should be no special configuration required for DataWedge, this tutorial is designed to use the default profile. If something does not work then please see the troubleshooting section at the end.

The application

All the application will be doing is sending Android Intents to the DataWedge service. DataWedge is a service on the device that contains a Broadcast receiver listening for a specific Intent action and determining what to do based on the extras contained in that Intent.

The UI will have 2 features:

1. Toggle the Scanner beam (i.e. simulate pressing the trigger in software). This is the " ScanSoftTrigger" API
2. Enable or Disable the scanner. This is the " Scanner Input Plugin" API.

I have omitted the code to create the 3 buttons on our Android UI

In our code, define some constants for the Intent actions and extras, these strings are detailed in the documentation for each API

```
 //  The Intent action is common for all DataWedge APIs
 private final static String DATAWEDGE_ACTION = "com.symbol.datawedge.api.ACTION";

 //  Toggle the scanner.  Define the intent extra key and value
 private final static String DATAWEDGE_EXTRA_KEY_SCANNER_TRIGGER_CONTROL = "com.symbol.datawedge.api.SOFT_SCAN_TRIGGER";
 private final static String DATAWEDGE_EXTRA_VALUE_TOGGLE_SCANNER = "TOGGLE_SCANNING";
```

```

//  Enable / Disable the scanner.  Define the intent extra key and values.
private final static String DATAWEDGE_EXTRA_KEY_SCANNER_CONTROL = "com.symbol.datawedge.api.SCANNER_INPUT_PLUGIN";
private final static String DATAWEDGE_EXTRA_VALUE_ENABLE_SCANNER = "ENABLE_PLUGIN";
private final static String DATAWEDGE_EXTRA_VALUE_DISABLE_SCANNER = "DISABLE_PLUGIN";
```

Then hook up the UI buttons in the onCreate method.

First the scanner beam toggle:

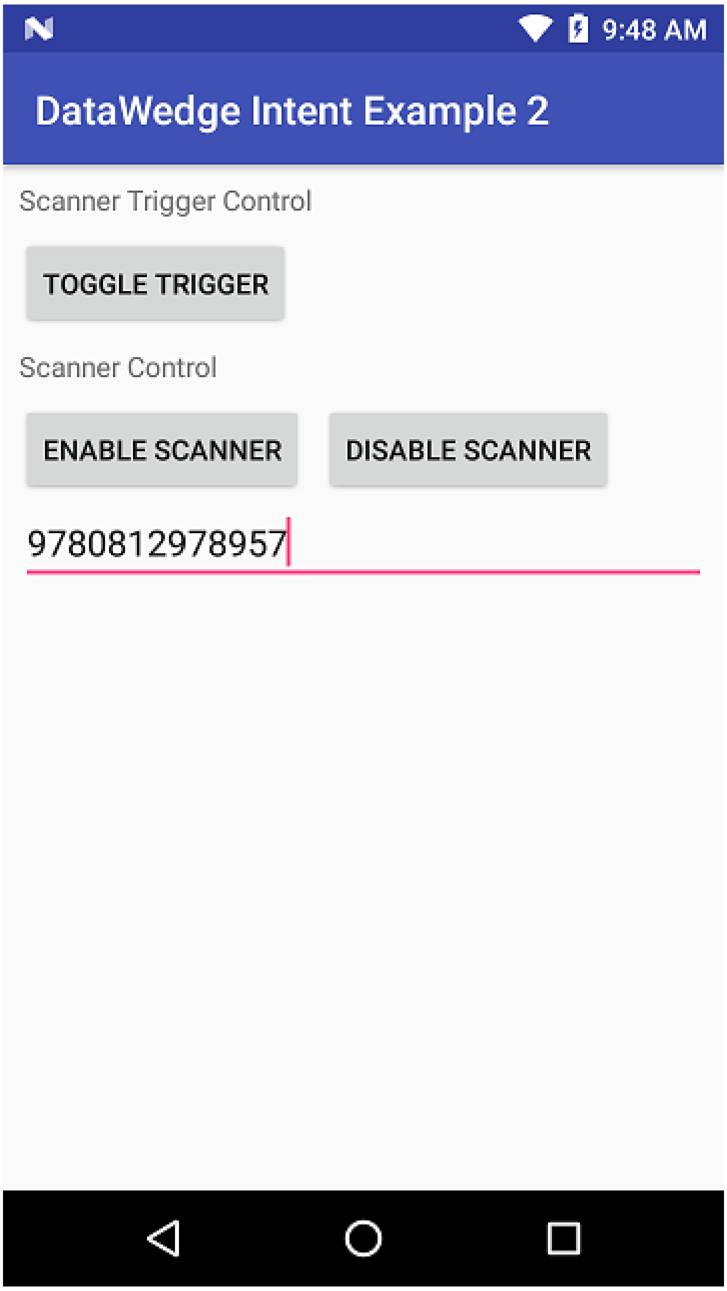
```
final Button btnToggleScan = (Button) findViewById(R.id.btnToggleScanner);
btnToggleScan.setOnClickListener(new View.OnClickListener() {
   public void onClick(View v) {
      Intent intent = new Intent();
      intent.setAction(DATAWEDGE_ACTION);
      intent.putExtra(DATAWEDGE_EXTRA_KEY_SCANNER_TRIGGER_CONTROL, DATAWEDGE_EXTRA_VALUE_TOGGLE_SCANNER);
      sendBroadcast(intent);
   }
});
```

Then the two buttons for enabling or disabling the scanner:

```
final Button btnEnableScanner = (Button) findViewById(R.id.btnEnableScanner);
btnEnableScanner.setOnClickListener(new View.OnClickListener() {
   public void onClick(View v) {
      Intent intent = new Intent();
      intent.setAction(DATAWEDGE_ACTION);
      intent.putExtra(DATAWEDGE_EXTRA_KEY_SCANNER_CONTROL, DATAWEDGE_EXTRA_VALUE_ENABLE_SCANNER);
      sendBroadcast(intent);
   }
});
final Button btnDisableScanner = (Button) findViewById(R.id.btnDisableScanner);
btnDisableScanner.setOnClickListener(new View.OnClickListener() {
   public void onClick(View v) {
      Intent intent = new Intent();
      intent.setAction(DATAWEDGE_ACTION);
      intent.putExtra(DATAWEDGE_EXTRA_KEY_SCANNER_CONTROL, DATAWEDGE_EXTRA_VALUE_DISABLE_SCANNER);
      sendBroadcast(intent);
   }
});
```

And that is really all you need to do to create an application which has basic control of the scanner using the DataWedge API. You can of course get a lot more complicated as the DW API has quite a few features

There is a sample application that accompanies this blog and shows all the code at https://github.com/darryncampbell/DataWedge-Intent-Example-2



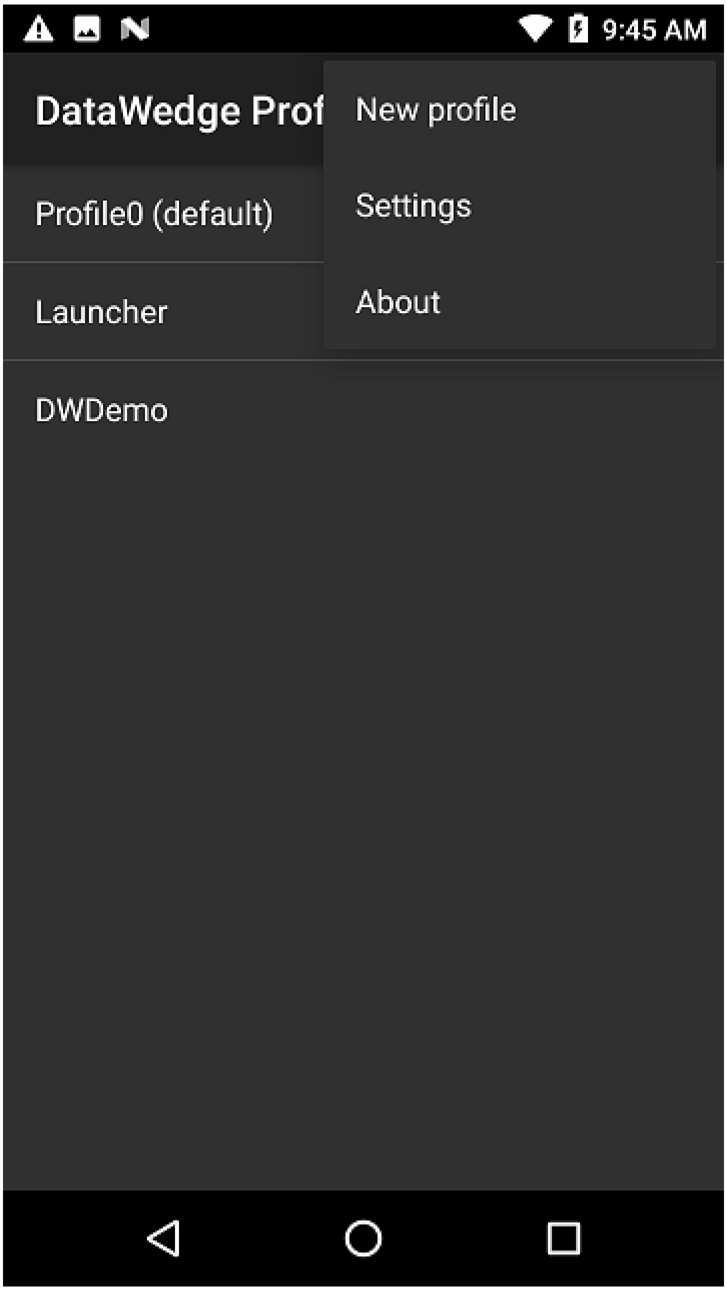
9780812978957

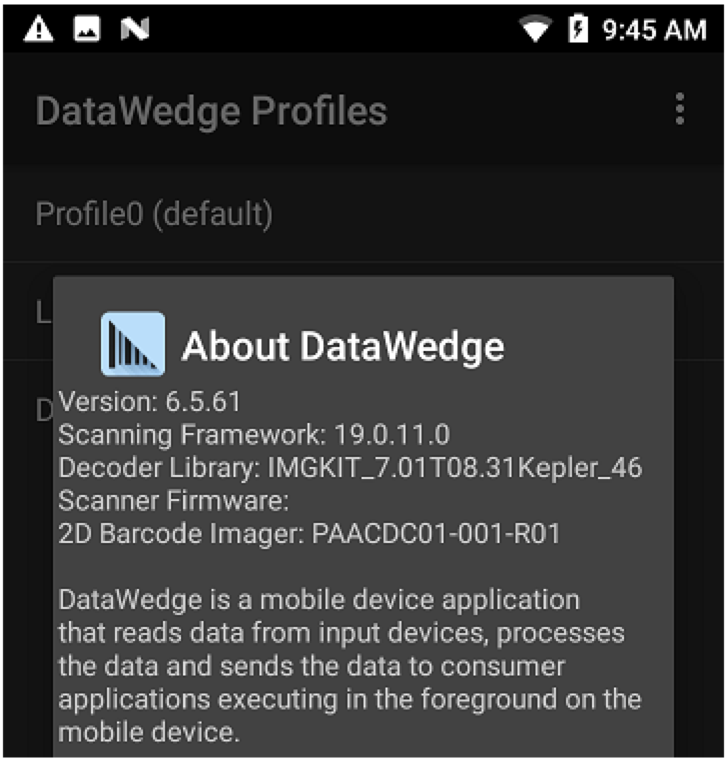
Troubleshooting:

If the application does not work for you then try some of these troubleshooting tips:

DataWedge Version

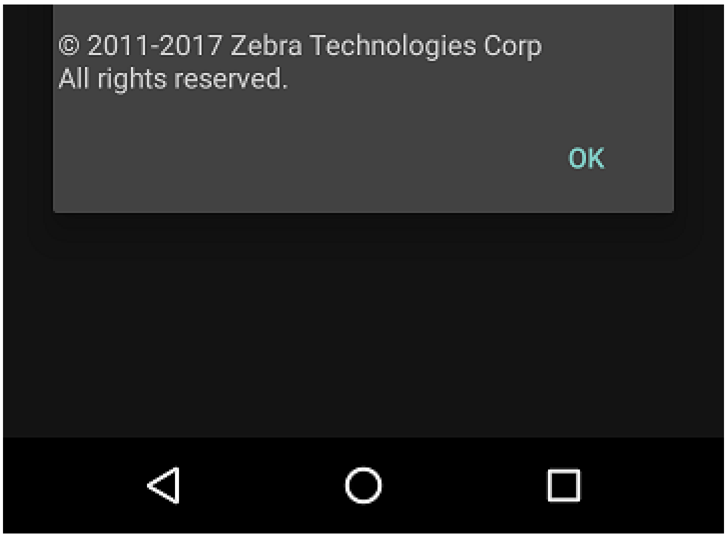
This application will only work on Zebra Android mobile devices running DataWedge 6.3 or higher. To check your datawedge version, launch the DataWedge application then go to the menu (3 dots), then 'about':





Version: 6.5.61
Scanning Framework: 19.0.11.0
Decoder Library: IMGKIT_7.01T08.31Kepler_46
Scanner Firmware:
2D Barcode Imager: PAACDC01-001-R01

DataWedge is a mobile device application that reads data from input devices, processes the data and sends the data to consumer applications executing in the foreground on the mobile device.

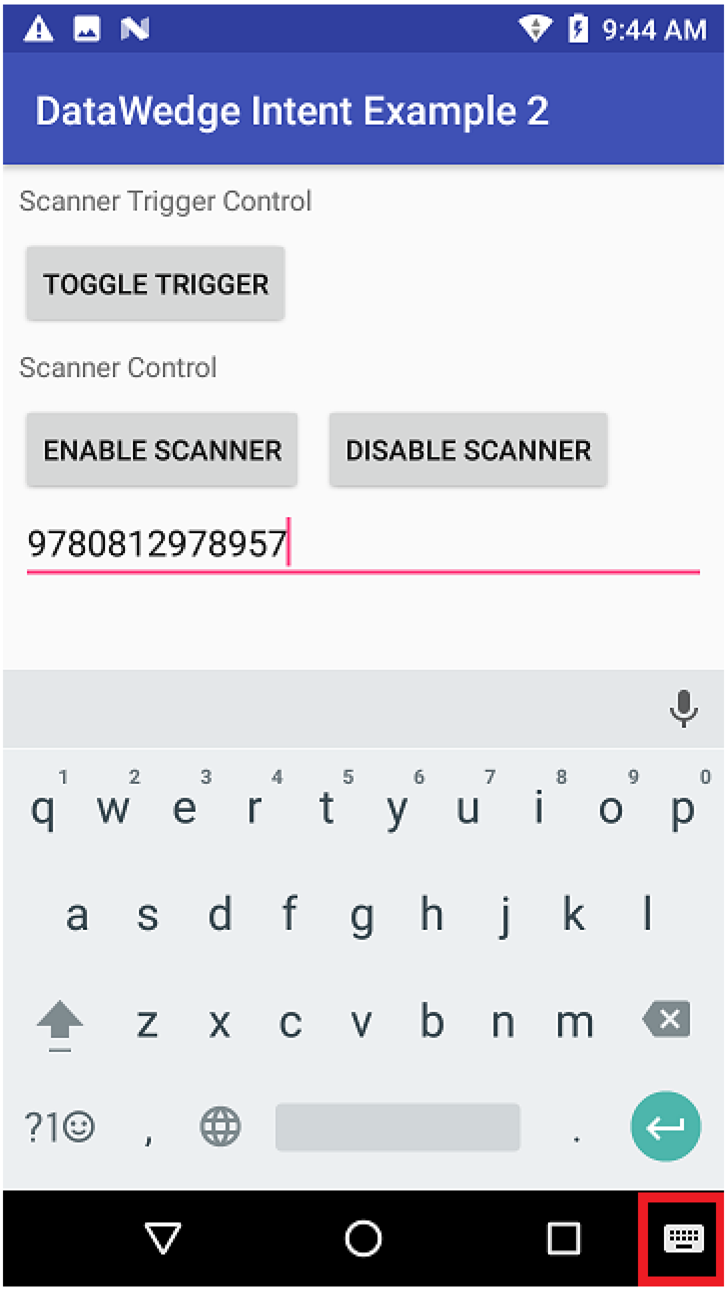


© 2011-2017 Zebra Technologies Corp
All rights reserved.

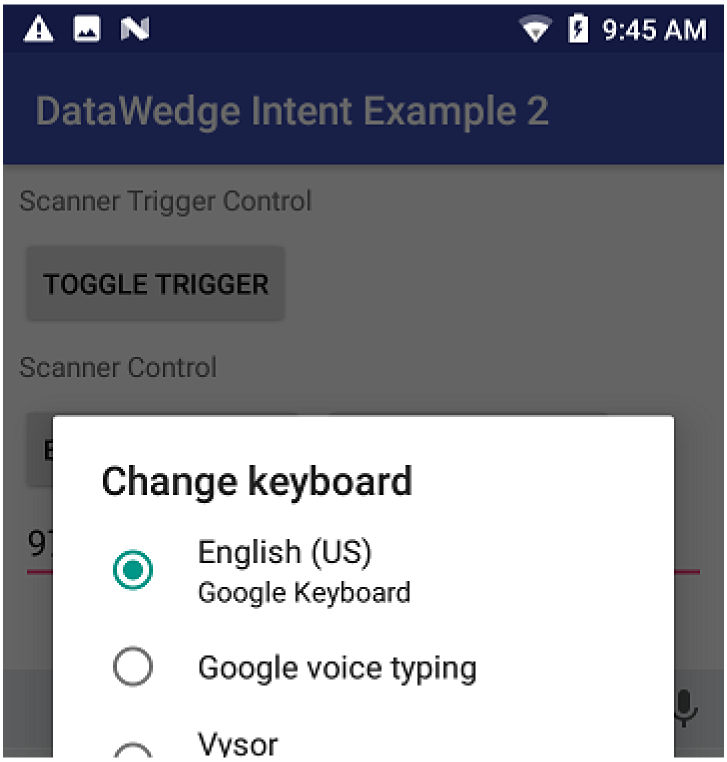
It is possible to use a version of the DataWedge APIs that work on 6.0 or higher, for the sake of simplicity I have used the more recent 6.3+ APIs in this example but you can see the 6.0+ APIs in my more fully featured project.

Vysor or other keyboard applications

This application depends on DataWedge sending keyboard output to the application after a successful scan. Vysor has been known to interrupt this workflow, it is recommended to uninstall Vysor or ensure a non-Vysor keyboard is selected (you may need to select the Vysor keyboard and then re-select the Google keyboard):



9780812978957

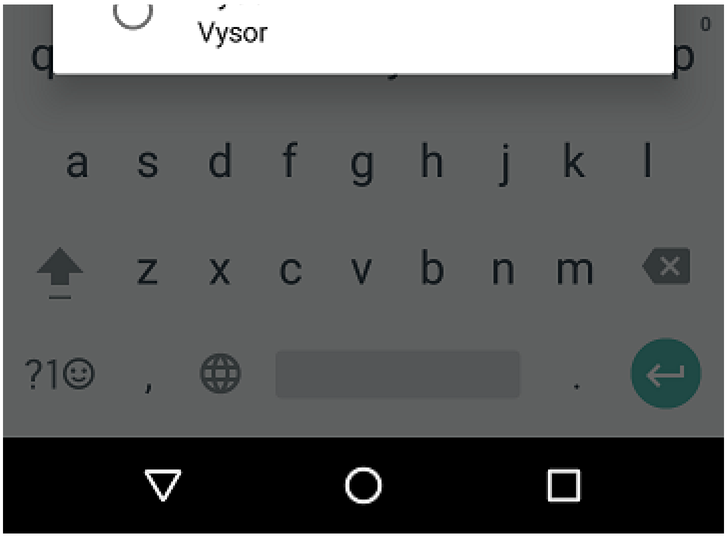


Change keyboard

● English (US)
Google Keyboard

○ Google voice typing

○ Vysor



DataWedge configuration

Unless you have changed any of the DataWedge configuration on the device, this application will use the default profile (Profile 0) with:

- Barcode input Enabled with common decoders
- Keystroke output Enabled

If you have changed any of these settings then you may find the application does not work e.g. disabling keystroke output or disabling the default profile.

There are multiple ways to resolve this including:

- You can restore DataWedge back to its default settings from the DataWedge application –> Menu (3 dots) –> Settings –> Restore
- You can create a new profile and associate it with this application

Share this:

Related

Tutorial: The DataWedge Intent API (with Xamarin)
10th April 2018
In "Zebra Technologies"

Tutorial: Scan with Datawedge Intent output on Zebra devices (with Xamarin)
16th January 2018
In "Zebra Technologies"

Tutorial: Scan with Datawedge Intent output on Zebra devices
13th December 2017
In "Zebra Technologies"

Category *Zebra Technologies*

Tags *Android* *DataWedge* *Tutorial*

14 Comments

Joachim Unger says:

19th July 2018 at 12:57 pm

Hello Darryn,

I am looking forward in developing a 1st Android scanner application for TC51 mobile computers.

In the past I programmed Intermec windows mobile devices. A part of these applications work with DataWedge without any programming and the other part – need for scanner enabling/disabling was developed with EMDK.

What is now the recommened way? The application has 10 activities with 1 to 2 scan fields. I want to be sure that the scanner can be safely controlled. My concern is the clean handover from activity to next activity and to get the data in the focused field what could be managed in WM.

Regards,
Jo

Reply

darryncampbell says:

19th July 2018 at 1:12 pm

Hi Jo, sounds like the easiest way to achieve that would be to use DataWedge, then you would not have to worry about managing the scanner between your 10 activities (since DataWedge profiles take care of that). DataWedge still supports a keystroke output (keywedge) so you can have the data inserted into the focused field but if you are rewriting your application I recommend using intent output (a scan is sent to your application as an intent) to give you more control over the scanned data

Reply

Nicat says:

22nd August 2019 at 3:40 pm

I looked at the code. But i did not found that where the result set to the textfield?

Reply

darryncampbell says:

23rd August 2019 at 7:37 am

There is no code to specifically enter the text into the textfield, the textfield needs to have focus and then scanned data will be put there by Datawedge as if it were being typed by a keyboard

Reply

Cesar Garcia says:

6th May 2020 at 11:55 am

Hi Darryn, I've developed some Android RFiD Apps for Zebra devices. Zebra strongly recomends the use of Datawedge for Scanning applications (that's my case, I work with both RFiD and Scan operations) but Datawedge is not fully working with RFiD, for example I can't get RSSi through intents (intent.getStringExtra(getResources().getString(R.string…), the use of memory banks (EPC, TID, etc) has not been correctly implemented, and there's no way to perform write tag operations. The other choice is using API3 but I'm not sure Zebra is keeping it up to date. If I want to continue working with Zebra MC33R what's the best choice?. Thanks a lot, regards.

Reply

darryncampbell says:

6th May 2020 at 12:08 pm

Hi Cesar,

Although as you say it is the case that Zebra are trying to encourage developers to use DataWedge for scanning over the EMDK (native scanning SDK) there is no such pressure for developers creating RFID applications. Exposing RFID through DataWedge is currently seen more as a convenience rather than the "recommended" development approach and if DataWedge does not work for you in capturing RFID data then please use the native API3 which is still fully supported. For some background on this, DataWedge, EMDK and the MC33 are owned by a separate business unit than the RFID reader and RFID API… whilst Zebra does everything it can to present a coherent solution for customers and developers, this kind of confusing messaging can sometimes occur.

Thanks
Darryn.

Reply

Ravi says:

13th May 2020 at 4:27 pm

Hi Darryn,
Could you please help me with the below scenario:
I want scanned RFID value from Zebra RFID scanner on the web page(in Textbox) of web application developed using .NET framework.
Could you please help me, how I can use DataWedge API's to achieve the above requirement or any other approaches will be appreciated. Can we access scanned RFID tag value on wep application built using .NET framework?
Awaiting for your response.
Thanks in advanced.

Reply

darryncampbell says:

13th May 2020 at 4:44 pm

You should be able to enable the Keystroke output plugin to receive the data in the textbox: https://techdocs.zebra.com/datawedge/7-6/guide/output/keystroke/. It probably makes sense to manually create your profile first with the DataWedge application and enable the keystroke plugin via the UI. Then you can enable it via the SET_CONFIG API once you know your app & DataWedge configuration is working.

Reply

Ravi says:

13th May 2020 at 6:56 pm

Thanks a lot Darryn for your quick response.
As per your response I need to create profile with DataWedge manually and then need to enable the keystroke plugin.
I am not clear with that API part, I mean how I can access that API into web application(built using .NET framework)?
Which API need to consume? Do you ave any sample code to do so? I have gone through the sample codes which are work around android intent and mostly code in java. How other platform applications can access these API's?
As I am very new to DataWedge please help me on this.
Could you please provide me more details on this?
Thanks in advance.

Reply

darryncampbell says:

14th May 2020 at 9:14 am

If you are using a web application then it may make the most sense to not use the API – just create a working profile manually and never modify it ☺ You may benefit from reading through the techdocs documentation for DataWedge, https://techdocs.zebra.com/datawedge/7-6/guide/gettingstarted/ and also the section on profiles https://techdocs.zebra.com/datawedge/7-6/guide/profiles/ particularly the RFID input plugin and Keystroke output plugin. Because you were asking about DataWedge I just assumed you were using an MC3300R – is that correct? If not then DataWedge will not work for you as that is the only RFID device DataWedge supports at the time of writing.

Reply

Ravi says:

14th May 2020 at 11:56 am

Thanks a lot again for your precious reply.
Yes I am using MC3300R scanner device.

César says:

25th May 2020 at 9:13 pm

I think last datawedge version for mc33 is 7.3.31
Depending on what features you use you can find some issues (sorry for my gramar).

Ravi says:

11th June 2020 at 1:26 pm

Hi Darryn, Thanks for helping me to get access to scanned data from RFID scanner to web application using DataWedge(Mentioned above). Now I am looking forward for second approach, what if I want to get access to scanned RFID data in Web Application using DataWedge API's. Is it possible to call API's using IavaScript(or JQuery AJAX etc) as same as used to call web API's? If yes can you provide some sample code to do so.

Thanks.

Reply

darryncampbell says:

11th June 2020 at 1:48 pm

Hi Ravi, there is no JavaScript API offered for DataWedge. There is a JavaScript API for RFID available through Enterprise Browser (https://techdocs.zebra.com/enterprise-browser/2-6/guide/about/) (https://techdocs.zebra.com/enterprise-browser/2-6/api/re2x/rfid/) but be aware that requires a separate license.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment