



# uFR EMV Java v1.0

# Table of contents

<b>About</b>	<b>3</b>
<b>FTDI drivers</b>	<b>3</b>
<b>Download uFR Library</b>	<b>3</b>
<b>Java uFCoder library usage</b>	<b>3</b>
<b>Java EMV Functions</b>	<b>4</b>
<b>Revision history</b>	<b>5</b>

## About

In this guide we will explain necessary steps for setting up uFCoder library and drivers for the usage of EMV functions in Java.

## FTDI drivers

uFCoder library uses FTDI drivers for communication with uFR series readers. Installation guides, depending on your platform, can be found here:

<https://www.ftdichip.com/Support/Documents/InstallGuides.htm>

If you encounter errors when opening communication with our uFR series readers, refer to this guide:

[https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/Reader\\_opening%20-%20possible\\_problems.pdf](https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/Reader_opening%20-%20possible_problems.pdf)

## Download uFR Library

1. Navigate to <https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-lib>
2. Download or clone urf-lib repository.
3. Depending on your platform and architecture, download appropriate .dll/.so/.dylib file. (For example, windows/x86\_64/uFCoder-x86\_64.dll)

## Java uFCoder library usage

To import native functions from the uFCoder library in your Java code, first, you will need to create an **interface** that will be extending **Library** class. For example, the following **uFRFunctions** interface:

```
422 public interface uFrFunctions extends Library {
423
424     String UFR_Status2String(int status);
425
426     String GetDllVersionStr();
427
428     int ReaderOpen();
429
430     int ReaderReset();
431
432     int ReaderClose();
433
434     int ReaderOpenEx(int reader_type, byte[] port_name, int port_interface, byte[] arg);
435
436     int EMV_GetPAN(String df_name, byte[] pan);
437
438     int EMV_GetLastTransaction(String df_name, byte[] last_transaction_info);
439
440 }
441
```

This interface will specify which functions the user will later call from the uFCoder library, and what parameter types those functions will require. Parameter types must be compatible with native types. To see native function parameter types, users can refer to our official "uFR Series NFC reader API" that can be found here: <https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc>.

After creating this interface, the next step is loading the library using Java's native methods. For example, by using the **Native.loadLibrary()** function in the following manner:

```
uFrFunctions ufr;  
  
public window() {  
    initialize();  
    try {  
        ufr = (uFrFunctions) Native.loadLibrary("uFCoder-x86_64.dll", uFrFunctions.class);  
    } catch (UnsatisfiedLinkError e) {  
        JOptionPane.showMessageDialog(null, "Unable to load library for specified path");  
    }  
}
```

After you've specified this path from which to load the library from, calling uFCoder library functions is done by using the variable **ufr**.

## Java EMV Functions

As of version 5.0.41 of uFCoder library, few particular EMV functions have been implemented in order to simplify usage of our uFR series NFC readers with EMV capable cards.

One of the functions we will demonstrate will be the **EMV\_GetPAN()** function used for extracting card PAN number.

```
int status = 0;
String df_name = "2PAY.SYS.DDF01";
// df_name for PSE1 system should be "1PAY.SYS.DDF01";

status = ufr.ReaderOpen();
if (status == 0) // 0 is UFR_OK status, refer to our API for status calls.
{
    status = ufr.SetISO14443_4_Mode();
    if (status == 0)
    {
        byte[] pan = new byte[128];
        status = ufr.EMV_GetPAN(df_name, pan);
        if (status == 0)
        {
            System.out.println(pan);
        } else
        {
            System.out.println("EMV_GetPAN() error occurred: " + ufr.UFR_Status2String(status));
        }
    } else
    {
        System.out.println("Error while switching into ISO 14443-3 mode: " + ufr.UFR_Status2String(status));
    }

    ufr.s_block_deselect((byte)100);
} else
{
    System.out.println("Reader opening error: " + ufr.UFR_Status2String(status));
}
```

As shown, by simply calling **ufr.EMV\_GetPAN()** and providing necessary parameters, users can quickly extract the PAN number from the EMV capable card.

For more details & references on API functions used in this short example, please see our "uFR Series NFC Reader API".

## Revision history

Date	Version	Comment
2019-10-31	1.0	Base document