

1- How Do I Add ProcessCamera View to My Activity's Design XML?

To add the ProcessCamera view to the XML of the activity under work, include the following tag:

```
<com.sana.logi.camera.library.ProcessCamera
    android:id="@+id/nfcReadCamera"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

2- How Do I Use the ProcessCamera Class?

- Create an instance of ProcessCamera inside the `onCreate()` method of the activity under work.
- Set the card type that is going to be scanned (i.e. passport, ID card, etc.)
- Set an instance of the `ScanResultInterface` to access the data obtained from the MRZ scan.
- Use `result()` method inside `ScanResultInterface` will parse and retrieve the data automatically.

An example use for setting an instance of `ScanResultInterface` and `result()` method can be seen on the following screenshot. Please note that `myView` represents an instance of `ProcessCamera` class.

```
1 myView = (ProcessCamera) findViewById(R.id.nfcReadCamera);
2 myView.setCardType(CardType.Passport);
3 myView.setScanResultInterface(new ScanResultInterface() {
    @Override
    4 public void result(final PassportModel passportModel) {
        Log.d( tag: "PassportModel", passportModel.toString());
    }
});
```

3- How Can I Perform a NFC Read to Retrieve a Document (Passports, ID Cards, etc.)

- **Step 1** – Use `NfcConnection` class to retrieve an instance of the NFC connection using `getInstance()` method. After getting the instance, use `init()` method to initialize the connection between the application, and the device's NFC chip. Please note that this should be declared inside `onCreate()` lifecycle method of the activity under work, before creating an instance of the `ProcessCamera` class. An example can be seen on the following screenshot.

```
1 NfcConnection.getInstance().init(getApplicationContext(), activity: this);  
  
myView = (ProcessCamera) findViewById(R.id.nfcreadCamera);
```

- **Step 2** – The following methods should be implemented in the Activity under work, in order to perform a NFC read.

```
@Override  
protected void onPause() {  
    NfcConnection.getInstance().onPause();  
    super.onPause();  
}  
  
@Override  
protected void onNewIntent(Intent intent) {  
    NfcConnection.getInstance().onNewIntent(intent);  
    super.onNewIntent(intent);  
}  
  
@Override  
protected void onResume() {  
    super.onResume();  
    NfcConnection.getInstance().onResume();  
}
```

- **Step 3** – To access the data obtained from the NFC read, an instance of `NfcInterface` must be set inside the `NfcConnection` class. 3 different methods: `results()`, `steps()` and `error()` will be provided by this interface. Results provides the data obtained after the NFC read. Steps provide information regarding the status of the NFC read process. Error provide information regarding any potential errors that are encountered during the NFC read process.

- An example for the implementation of these methods can be observed on the following screenshot:

```
1 NfcConnection.getInstance().setNfcInterface(new NfcInterface() {
    @Override
2 public void result(PassportModel nfcData) {
    Log.d(TAG, msg: "result: "+nfcData.toString());
    }

    @Override
3 public void steps(String file, String status) {
    Log.d(TAG, msg: "steps: "+ file+" "+ status);
    }

    @Override
4 public void error(Exception ex, String message) {
    Log.d(TAG, msg: "error: "+ex.getMessage());
    }
}
```

Important: To perform a NFC read, the `PassportModel` object obtained from MRZ scan should be passed on to the `NfcConnection` instance mentioned of Step 1.

```
try {
    NfcConnection.getInstance().setPassportModel(passportModel);
} catch (Exception ex){
    ex.printStackTrace();
}
```

A complete version of the implementation for MRZ scan and NFC read can be seen on the screenshots bellow:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    NfcConnection.getInstance().init(getApplicationContext(), activity: this);

    ProcessCamera myView = (ProcessCamera) findViewById(R.id.nfcreadCamera);
    myView.setCardType(CardType.Passport);
    myView.setScanResultInterface(new ScanResultInterface() {
        @Override
        public void result(final PassportModel passportModel) {
            Log.d(tag: "PassportModel", passportModel.toString());
            try {
                NfcConnection.getInstance().setPassportModel(passportModel);
            } catch (Exception ex){
                ex.printStackTrace();
            }
        }
    });
}
```

```
nfcbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            NfcConnection.getInstance().setNfcInterface(new NfcInterface() {
                @Override
                public void result(PassportModel nfcData) {
                    Log.d(TAG, msg: "result: "+nfcData.toString());
                }

                @Override
                public void steps(String file, String status) {
                    Log.d(TAG, msg: "steps: "+ file+" "+ status);
                }

                @Override
                public void error(Exception ex, String message) {
                    Log.d(TAG, msg: "error: "+ex.getMessage());
                }
            });
        } catch (Exception ex){
            Log.d(TAG, msg: "NfcConnection: "+ex.getMessage());
        }
    }
});
```