

# Biometric signature recording scenario on POS with a native app on a tablet - Example of a car selling scenario

## The biometric signature story v 3.2

Alice and Frank want to buy a new car. So they are going to the local car dealer to look and check the models to buy one. The car dealer is using eSignAnywhere for digital signing. To record biometric signatures on the point-of-sales (POS), a native app (e.g. SIGNificant App or SignAnyWhere App on iOS, Android or Windows ) is required on the tablet device.

You find a [list of available stories here](#). Please also see the developer mode for this tutorial here: [Visit the developer mode](#).



## Use Case Description

1. The contract is prepared via salesman on his computer. He enters the data, discount, payment methods and additional data
2. The salesman opens the contract on the tablet pc via
  - a. Option (1): Email – an email with the link to the document is sent to the tablet. The salesman clicks on the link to open the document in the SIGNificant app for signing.
  - b. Option (2): QR Code – a QR code is shown on the computer of the salesman and scans it via tablet to open the document in the SIGNificant app
  - c. Option (3): a company notification system is used to push the link for the document directly to the tablet
3. Alice and Frank are signing in the native app on the tablet (in this example via biometric signature)
4. The document is finished and synchronized with the server (via salesman)
5. The computer of the salesman shows the finished document and Frank receives a copy of the contract via Email. Optionally the salesman can now print a carbon copy for Alice and Frank

### Information

This example uses the native SIGNificant App (for Android and iOS) and requires an eSignAnywhere Instance with an open Workstep Controller Process interface. This is available at <https://demo.esignanywhere.net/>, so it is recommended to create an free account at <https://demo.esignanywhere.net/> for testing this example or use your own on-premise or private-SaaS eSignAnywhere instance.

Note: Due the security limitations, this example will not work on <http://significant.com>. Please use <https://demo.esignanywhere.net/> instead. The following example and description uses the endpoints of <https://demo.esignanywhere.net/>.

## Integration

The integration of this use case requires eSignAnywhere and the SIGNificant App on the tablet. Due the integration of the native App, this integration is a little bit different from eSignAnywhere-only integrations.

The creation of the workflow is similar, except that in one step (so one recipient) two persons are signing at once ("hosted meeting"). This is done via the use of the native SIGNificant App. Therefore no authentication is required. Moreover, this scenario is perfect for biometric signatures (advanced electronic signatures) and due the biometric signatures it doesn't require any authentication.

To bring the document to the tablet there are some **options** available:

1. **Email:** a email is sent to the tablet and opened by the salesman. The email contains a link, which opens automatically the app and the document for signing.
2. **QR Code:** on the computer of the salesman is a QR code shown, after the creation of the workflow. The salesman is opening the SIGNificant App and scans the QR code. The app opens automatically the document for signing. In this option there is no email sent in the first signing step.
3. **Custom Notification:** a custom notification is used to push the link to the tablet. In that case an even deeper integration (via SIGNificant App SDK) can be used. This means on the tablet is specific car dealership app installed, which has integrated the SIGNificant App SDK for signing. This specific car dealership has integrated the notification service and gets the information for signing task and automatically opens the document for signing.

This example shows two integration for option 1 (Email) and option 2 (QR Code).

## Example with Option (1) Email

The workflow is the following:

Endpoint (URI) for REST: <https://demo.esignanywhere.net/Api>

1. Create the PDF based on the data provided by the salesman in the car dealerships CRM
2. Upload the PDF to eSignAnywhere
3. Prepare the envelope and set CC recipient
4. Send Envelope
5. Open the tablets email inbox and click on the link of the eSignAnywhere email
6. Frank and Alice are signing & finishing the document in the App
7. Frank receives and copy of the document via Email
8. The `callback` is used to update the car dealerships CRM

```
{
  "Documents": [{
    "FileId": "c30dbbfc-1234-1234-ae0f-55ee351d9452",
    "DocumentNumber": 1
  }
],
  "Name": "Test",
  "Activities": [{
    "Action": {
      "Sign": {
        "RecipientConfiguration": {
          "IncludedEmailAppLinks": {
            "Android": true,
            "iOS": true,
            "Windows": true
          },
          "ContactInformation": {
            "Email": "janedoe@sample.com",
            "GivenName": "Jane",
            "Surname": "Doe",
            "LanguageCode": "EN"
          }
        }
      },
      "Elements": {
        "Signatures": [{
```

```

"TaskConfiguration": {
  "OrderDefinition": {
    "OrderIndex": 0
  }
},
"ElementId": "sample sig biometric",
"Required": true,
"DocumentNumber": 1,
"DisplayName": "Sign here",
"AllowedSignatureTypes": {
  "Biometric": {
    "BiometricVerification": false,
    "AllowBiometricStoringOnly": false,
    "StoreSignedResponseWithoutBioData": true,
    "AllowSkipBiometricVerification": false,
    "BiometricServerUserId": "string",
    "SignaturePositioning": "WithinField",
    "Preferred": true,
    "StampImprintConfiguration": {
      "DisplayExtraInformation": true,
      "DisplayEmail": true,
      "DisplayTransactionId": true,
      "DisplayTransactionToken": true,
      "DisplayPhoneNumber": true,
      "UseCustomStampImprint": true,
      "DisplayIp": true,
      "DisplayName": true,
      "DisplaySignatureDate": true,
      "FontName": "string",
      "FontSizeInPt": 0
    }
  }
},
"FieldDefinition": {
  "Position": {
    "PageNumber": 1,
    "X": 100,
    "Y": 200
  },
  "Size": {
    "Width": 100,
    "Height": 70
  }
}
],
"SigningGroup": "firstSigner"
}, {
  "Action": {
    "SendCopy": {
      "RecipientConfiguration": {
        "ContactInformation": {
          "Email": "janedoe@sample.com",
          "GivenName": "John",
          "Surname": "Doe",
          "LanguageCode": "EN"
        }
      }
    }
  }
}
]
}

```

## Example with Option (2) QR Code

Workflow:

1. Create the PDF based on the data provided by the salesman in the car dealerships CRM
2. Upload the PDF to eSignAnywhere
3. Prepare the envelope and set CC recipient
4. Send Envelope (with disableEmail=true for first recipient)
5. The returned envelopeID is used to call <https://demo.esignanywhere.net/api/v6/envelope/{envelopeId}/viewerlinks> to retrieve the redirectionURL. This URL forwards to a HTTP forward to the SignAnywhere Viewer (Web-Client), but we need the workstepId for the App. Therefore you have to add to the WorkstepRedirectionUrl an additional parameter to get it directly (&responseType=returnWorkstepId).
6. The car dealership CRM shows the QR Code for the signing task. The salesman scans the QR Code from the display via SIGNificant App to open the document
7. Frank and Alice are signing & finishing the document in the App
8. Frank receives and copy of the document via Email
9. The callback is used to update the car dealerships CRM

### Retrieve WorkstepID and generate the QR Code

The first recipient in this example is the tablet. The SIGNificant App requires the workstepId for signing the document. After you created the envelope (SendEnvelope) you do the following:

1. Call <https://demo.esignanywhere.net/api/v6/envelope/{envelopeId}/viewerlinks> to retrieve the redirectionURL.
2. The response contains the workstepRedirectionUrl. This URL is the link to the SignAnywhere Viewer (Web-Client), which forwards to the eSAW redirector and delivers a link with a workstepId parameter. With the additional parameter&responseType=returnWorkstepId you directly receive the workstepId:

"<https://demo.esignanywhere.net/workstepredirector/sign?identifier=1234~E8/79IYvRlwR4wl41234hbX6NHKhxuKv~jlsG4trySqTsn12345JuSJm/NJo8Mmii7G~1234==&responseType=returnWorkstepId>

3. Extract the workstepID from the response.
4. Prepare Data for QR Code: the QR Code contains a link to the server and the workstepId (Signing-Task). So you have to use the extracted workstepId to build a link (replace ##WORKSTEPID##):  
<http://launch.xyzmo.com/SignificantAndroidAppLauncher.aspx?WorkstepId=##WORKSTEPID##&server=demo.esignanywhere.net/WSCProcess&protocol=https>
5. Generate a QR Code with that link (e.g. with a proper QR Code library, as example you may use a offline QR Code Generator [<https://sites.google.com/site/qrcodeforwn/home/qr-code-generator-for-wireless-networks>]).
6. Show the QR Code and scan it with the SIGNificant App (Button lower left above settings symbol)