

iOS NPRFramework Documentation

To integrate NPRSDK in your iOS app please follow below steps:

1.1. Integration

1. To link SDK and supporting frameworks from 'Supporting Frameworks' folder(can be found inside sample app):Click on Project > Target > General Tab > Frameworks, Libraries and Embedded contents > Click on plus button > Add NPRFramework.xcframework. Follow same procedure for ReadID, ReadID_UI and Facetec frameworks. These frameworks are needed for NPRFRAMEWORK to work without any dependency errors.
2. Go to 'copy bundle resources' of your app and import readid_certificates.plist, images, fonts, storyboardc from NPRFramework.xcframework > iOS-arm64 > NPRFramework folder.
3. Check NtdrSDK colors in Assets file of sample project, and copy-paste same in your app. Your app can add their own colors under same name.
4. Add "import NPRFramework" line at the top of view controller where you wish to consume SDK functionalities.

1.2. SDK usage Prepare SDK and response instance in your viewDidLoad method like below:

```
1. //SDK Object which helps to access SDK functionality
var api: NtdrAPI?

2. //Response object return by SDK functions
var ntdrResponse: NtdrResponse?

3. //Configuration class's object which holds all the configuration, you need to pass to SDK
let config = AcquisitionConfig(enviroment: .PREPROD, ntdr_app_doclist: "", ntdr_expire_grace: "", ntdr_facematch_min: "", ntdr_doc_issuers: "", ntdr_images: "")

4. //Initialize the SDK with valid key and version
self.api = NtdrSDK.load(key: "your_sdk_key", version: NtdrSDK.VERSION)

5. //Provide ViewController where SDK will perform UI actions
api?.setContext(viewController: self)
```

1.3. SDK CUSTOMIZATIONS

NtdrSDK Theme Object: Document Scan

```
api?.sdkTheme?.selectIDTitleText = "Select ID type"
api?.sdkTheme?.selectIDDescriptionText = "Your identification document must have a biometric chip. If in doubt, choose passport. "
api?.sdkTheme?.selectIDPassportOptionText = " Passport"
api?.sdkTheme?.selectIDResidentCardOptionText = " Residence Card"
api?.sdkTheme?.selectIDNextButtonTitle = "Next"
```

Activation Code Screen Buttons Customizations

```
api?.sdkTheme?.codeScreenScanQRButtonTitleColor (default color is same as previous versions UIColor.white)
api?.sdkTheme?.codeScreenScanQRButtonBackgroundColor (default color is same as previous versions Ntdr.Button.BackgroundColor)
api?.sdkTheme?.qrScanScreenEnterManuallyButtonTitleColor (default color is same as previous versions UIColor.white)
api?.sdkTheme?.qrScanScreenEnterManuallyButtonBackgroundColor (default color is same as previous versions Ntdr.Button.BackgroundColor)
```

Select ID Type:

When document types are configured from server side then Select ID Type screen will be visible and you can do following customizations of it.

You can include your own images for icons shown in front of the document option text. You can either include images provided with SDK or your own image icons with names like "documentOptionOneImage", "documentOptionTwoImage", "documentOptionThreeImage".

Same goes for arrow icon shown in front of the document options with name "detailArrow".

1. SDK integration and usage

```
//Activation code textfield has two modes now
// .defaultStyle and .boxedStyle
//By default it's set to boxedStyle
You can show default plain textfield using following statement
api?.sdkTheme?.codeTextFieldStyle = .defaultStyle
```

Document Scan Flow Font Customisation

```
ThemeCustomization.passportScanViewCustomization()//access the properties for document scan customizations.
//ThemeCustomization.passportScanViewCustomization.buttonFont =

//ThemeCustomization.passportScanViewCustomization.titleFont =

//ThemeCustomization.passportScanViewCustomization.textFont =
```

General SDK Screens Button Appearance Customisation

```
//api?.sdkTheme?.primaryButtonCornerRadius =

//api?.sdkTheme?.primaryButtonBorderColor =

//api?.sdkTheme?.primaryButtonBorderWidth =
```

General SDK Font Customisation

```
//api?.sdkTheme?.primaryButtonFont =

//api?.sdkTheme?.screenTitleFont =

//api?.sdkTheme?.descriptionInfoFont =
```

NtdrSDK Close Button: It is possible to show close button on each screen of NPRFramework. Document Scan and FaceScan flows are not considered for this customizations as they have their own screens and close buttons. Following property shows close button. By default this property is true that means you won't be able to see close button unless you set false to below property. `api?.sdkTheme?.nprCloseButtonAppearance.closeButtonIsHidden = false` If you want to screen wise show or hide this button then you can set true or false before each process. The close button appearing on the screen is further customizable with following properties: Like you set the `closeButtonIsHidden` property, you can set below properties as well: `closeButtonIsHidden`

```
closeButtonImage
closeButtonBackGroundImage
closeButtonTitle
closeButtonBackgroundColor
closeButtonTitleColor
closeButtonWidth = 75.0
closeButtonHeight = 35.0
closeButtonCornerRadius
```

NtdrSDK Theme Object: Face Verification

```
api?.sdkTheme?.faceScanIntroTitle = "Take a selfie "

api?.sdkTheme?.faceScanIntroDescription = "Face recognition is performed to verify your identity. This is to make sure that you are the legitimate document owner. "

api?.sdkTheme?.faceScanIntroNextButtonTitleColor = .blue

api?.sdkTheme?.faceScanIntroNextButtonBackgroundColor = .white

api?.sdkTheme?.faceScanIntroNextButtonTitle = "Next"

api?.sdkTheme?.faceVerificationUICustomization.

api?.sdkTheme?.faceScanSuccessMessageText = NSLocalizedString("FaceScan_SuccessMessage", comment: "")

api?.sdkTheme?.faceScanUploadMessage = NSLocalizedString("FaceScan_UploadMessage", comment: "")
```

1. SDK integration and usage

`api?.sdkTheme?.allowNFCSkipReadingAfterAttempts = -1` //To hide skip button on NFC flow. (Or you can give positive number to allow NFC Skip reading after that many attempts)

NtdrSDK Colors:

Now it's possible to change color of screen title and description info of activation code, select ID and Liveness check intro screens with a property accessor:

`api?.sdkTheme?.screenTitleColor` (default color is same as previous versions `Ntdr.HeaderTextColor`)
`api?.sdkTheme?.screenDescriptionInfoTextColor` (default color is same as previous versions `Ntdr.Black`)

Check NtdrSDK colors in Assets file of sample project, and copy & paste same in your app. Your app can add their own colors under same name

Color Keys should be kept same. Below are the keys to which you should map colors in Assets file of your project. (Follow Sample App's Asset file for more clarity)

Following are the color keys:

`Ntdr.Black`

`Ntdr.Button.BackgroundColor`

`Ntdr.Button.TitleTextColor`

`Ntdr.HeaderTextColor`

`Ntdr.PrimaryColor`

`Ntdr.Progress.BackgroundColor`

`Ntdr.ProgressBar.EmptyColor`

`Ntdr.ProgressBar.FillColor`

`Ntdr.SecondaryColor`

`Ntdr.SubHeaderTextColor`

`Ntdr.Textfield.PlaceHolderColor`

`Ntdr.TextFieled.BorderColor`

`Ntdr.White`

Document Scan Colors

`readid_background_color`

`readid_icon_button_color`

`readid_icon_button_disabled_color`

`readid_icon_button_pressed_color`

`readid_loading_color`

`readid_passport_color`

`readid_passport_inner_page_color`

`readid_passport_inner_page_content_color`

`readid_passport_symbol_color`

`readid_primary_button_border_color`

`readid_primary_button_color`

`readid_primary_button_disabled_border_color`

`readid_primary_button_disabled_color`

`readid_primary_button_disabled_text_color`

`readid_primary_button_pressed_border_color`

`readid_primary_button_pressed_color`

`readid_section_header_background_color`

`readid_section_header_text_color`

2. SDK Operations

1.4.

Consuming SDK features Upon successful initialisation, an instance of the passport reader implementation is returned. The api implementation is henceforth used to request the sdk to perform authentication tasks.

The API implementation has the following methods:

Parameter	Description
api.init(..)	(For internal use only)
api.join(..)	Attach the SDK to an already existing authentication request
api.acquire(..)	Request the SDK to perform bio-metric data collection using any of the supported acquisition methods
api.update(..)	Inform the server to update the status of the authentication status (e.g. cancel or verify)
api.upload(..)	Post extra user data to the server
api.status()	Query the server for the current session state
api.commit(..)	Commit an authentication request (only for backchannel authentication)

A session join request enables the device app to attach to an existing Passport Reader session. Session join requires a valid activation code. The activation code is primarily generated by the Passport Reader back-end and displayed to the user as a 6-digit code with accompanying QR. The SDK can be requested to prompt the user to either enter the code, or scan the QR.

- **Acquire activation code**

```
NtdrResponse acquireCode = api.acquire(AcquisitionObject.ACTIVATION_CODE, config);
String activationCode = acquireCode.get(NtdrResponseKeys.INIT_CODE);
```

Alternatively, the code may already be available to the calling application by other means, such as deep linking parameters or from a remote server that the app trusts (backchannel authentication).
The activation code can now be used to attach to an existing session.

- **Session join**

```
Map/dictionary config = new Map();
config.set(AcquisitionConfigKeys.ENVIRONMENT, SDKEnv.PREPROD);
NtdrResponse sessionJoin = api.join(activationCode, config);
String tid = sessionJoin.get(NtdrResponseKeys.TRANSACTION_ID);
```

The session join, in addition to attaching to an existing session, also returns context specific configurations such as server timeouts, supported operations, and authentication contexts. The SDK will store the response data for the entire period of the authentication transaction.
The sample code above is configured for joining sessions in the Nets preproduction environment (customer test).
After a successful session join, the application is able to perform authentication operations and bio-metric data collection.
Note that the SDK can be used to perform transaction in all supported server environments. For development and testing, app developers should use the appropriate SDKEnv key.
For production (live apps) the SDKEnv.PROD key must be provided as environment configuration.

2.1. Collecting biometric data

With a transaction ID, it is now possible to request the SDK to retrieve bio-metric data using any of the supported mobile device sensors.

The SDK provides an acquire() method for every supported sensor.

```
NtdrResponse scanResponse = api.acquire(AcquisitionObject.NFC_SCAN, config);
```

The scanned NFC data in the code sample above is persisted on the server side. The data will be used during authentication verification when the session is committed.

The following table lists all supported acquisition objects that can be queried for data.

Data source	Sensor	Description
NFC chip	NFC	Scan machine readable travel document data stored in the chip. In addition to general information such as name, issuing country and expiration data, the NFC chip may also contain other details such as date and place of birth, nationality, gender, etc. The NFC data also contains an ID photo of the document holder. NtdrResponse scanNFCResponse = api.acquire(AcquisitionObject.NFC_SCAN, config);
3D facemap	Front facing camera	Scan user face for facial recognition purposes The SDK provides an acquire() method for every supported sensor. NtdrResponse scanFacemapResponse = api.acquire(AcquisitionObject.FACEMAP_SCAN, config);

2. SDK Operations

MRZ image	Camera	Take a photo of the travel document MRZ page The SDK provides an acquire() method for every supported sensor. NtdrResponse scanMRZResponse = api.acquire(AcquisitionObject.MRZ_SCAN, config);
Selfie image	Front facing camera	Take a standard photo selfie using the front facing camera The SDK provides an acquire() method for every supported sensor. NtdrResponse selfieResponse = api.acquire(AcquisitionObject.SELFIE_IMAGE, config);
Photo	Camera	Take a generic photo using any available camera The SDK provides an acquire() method for every supported sensor. NtdrResponse documentPhotoResponse = api.acquire(AcquisitionObject.DOCUMENT_MAIN_IMAGE, config);

2.2. Uploading custom biometric data

The SDK allows the calling application to add other externally acquired biodata to the server. The biodata must be in a format specified by the server.

- **Custom biodata**

```
NtdrResponse uploadResponse = api.upload(AcquisitionObject.SELFIE_IMAGE, String data);
```

The server will respond with either a pass, or a fail response.

2.3. Updating session status

Before each acquisition operation, the server status of the authentication session must be updated.

```
NtdrResponse updateResponse = api.update(action, message);
```

Action	Message	Description
update	one of: "wait-for-biodata", "wait-for-facemap", "wait-for-verify"	update session state: wait-for-biodata (expect biodata next), wait-for-facemap (expect selfie facemap next), wait-for-verify (all data acquired, start verification)
cancel	message text	optional text, reason for cancelling
timeout	message text	optional text

2.4. Finalising the authentication session

After all bio-data has been collected, the session is ready for verification. Verification will check that the collected bio-data validates and that session parameters have been fulfilled.

To finalise a session, the session state must first be updated to "wait-for-verify". This informs the server that no new data will be collected.

- **Session finalisation**

```
NtdrResponse updateResponse = api.update("update", "wait-for-verify");
```

If the authentication session was initiated in a web browser, the passport reader web page will automatically proceed to the authentication verification results page. At this point, the authentication transaction is complete. No further operations can be requested from the server.

If the authentication session was initiated through the back channel (no web UI), the session must be committed to trigger authentication verification.

- **Session commit**

```
NtdrResponse commitResponse = api.commit();
```

The commit response will indicate the result of the authentication verification.

2.5. Session configuration

Some operation that the API provides take a configuration object which can be used to set up various runtime parameters. The following table lists all parameters defined by AcquisitionConfigKeys:

Key	Description
ENVIRONMENT	Used to configure the service environment to communicate with. This key is used by the join operation.
NTDR_APP_DOCLIST	(not implemented)

2. SDK Operations

NTDR_EXPIRE_GRACE	This parameter determines which document expiry dates to allow when authenticating the user. Example: "30" means to allow documents whose expiry dates have not exceeded 30 days. "-90" will check that there still are at least 90 days left before the document expires Not supported in this version
NTDR_FACEMATCH_MIN	Specifies the minimum face match level required for successful authentication. Integer value ranges from 1 to 8. 1 is used if this they is not configured. Not supported in this version
NTDR_DOC_ISSUERS	Comma-separated list of issuing country codes whose documents will be accepted during the authentication session. Example: "UTO" will only allow documents issued by Utopia. Not supported in this version
NTDR_IMAGES	(not implemented)

2.6. SDK Success & Error handling:

Every public method in the SDK returns the following type inside the completion block.
"Result<IdvResult, ResponseCode>"

IdvResult is the success parameter and the ResponseCode is for failure.

Example of handling:

```
self.api?.update(action: "update", message: "wait-for-facemap", response: { facemapWaitResult in
    switch facemapWaitResult {
    case .success(let facemapWaitSuccess):
        break
    case .failure(let responseCode):

        switch responseCode {
            //Different error cases can be handled as intended by the calling app.
        }
        break
    }
})
```

The responseCode is the object of the enum ResponseCode which contains different error cases that can occur inside the SDK.
Following section explains all the ResponseCode.

ResponseCode has three parameters to understand the errors in a better way.
(code, message, genericMessage)

2.7. 2. Errors: Errors and their codes:

```
case .STATUS_INVALID_ACTIVATION_CODE:
    return 1001
case .STATUS_INVALID_DEVICE_CODE:
    return 1002
case .STATUS_BAD_BIODATA:
    return 1003
case .STATUS_BAD_BIODATA_DOCTYPE:
    return 1004
case .STATUS_BAD_BIODATA_EXPIRY:
    return 1005
case .STATUS_BAD_BIODATA_ISSUER:
    return 1006
case .STATUS_BAD_FACEMAP:
    return 1007
case .STATUS_BAD_PROTOCOL:
    return 1008
case .STATUS_INVALID_DEVICE_REQUEST:
    return 1009
case .STATUS_INVALID_SESSION:
    return 1010
case .STATUS_AUTH_FAILED:
    return 1011
case .UNKNOWN_QR_CODE:
    return 1111
case .UNKNOWN_ERROR:
    return 1112
case .NETWORK_ERROR:
    return 1113
case .RETRY_LIMIT_REACHED:
    return 1114
case .FACETEC_INITIALIZATION_FAILED:
    return 1115
```

2. SDK Operations

```
case .INVALID_JOIN_SESSION:
    return 1116
case .INVALID_UPDATE_SESSION:
    return 1117
case .INVALID_COMMIT_SESSION:
    return 1118
case .INVALID_STATUS_SESSION:
    return 1119
case .OCR_SCAN_FAILED:
    return 1120
case .BIODATA_UPLOAD_FAILED:
    return 1121
case .PARSING_ERROR:
    return 1122
case .NO_RESPONSE:
    return 1123
case .UNSECURE_SERVER_PINNING_FAILED:
    return 1124
case .EMPTY_ACCESS_KEY:
    return 1125
case .EMPTY_BIODATA_PROXY:
    return 1126
case .INVALID_BIODATA_ENDPOINT:
    return 1127
case .READID_INITIALIZATION_FAILED:
    return 1128
case .INVALID_FACEMAP_TOKEN:
    return 1129
case .FACESCAN_NO_SESSION_RESULT:
    return 1130
case .CANCELLED_BY_USER:
    return 1133
case .NO_CHIP:
    return 2001
case .CONNECTION_ERROR:
    return 2002
case .USER_CANCELLED:
    return 2003
case .CAMERA_PERMISSION_DENIED:
    return 2004
case .WRONG_DOCUMENT_TYPE:
    return 2005
case .TOO_MANY_REQUESTS:
    return 2006
case .SERVER_ERROR:
    return 2007
case .AUTHORISATION_ERROR:
    return 2008
case .SERVER_SESSION_ERROR:
    return 2009
case .SSL_ERROR:
    return 2010
case .SESSION_EXPIRED:
    return 2011
case .NFC_ACCESS_CONTROL_ERROR:
    return 2012
case .BACKGROUND_TIMEOUT:
    return 2016
case .NFC_ERROR:
    return 2017
case .UNSUPPORTED_DOCUMENT:
    return 2019
case .NFC_NOT_SUPPORTED:
    return 2020
case .EXPIRED_SDK:
    return 2021
case .BACK_NAVIGATION:
    return 2022
case .INVALID_CONFIGURATION:
    return 2023
case .VIZ_CAPTURE_TIMEOUT:
    return 2024
case .NON_PRODUCTION_MODE_KEY_INVALID:
    return 3001
case .NON_PRODUCTION_MODE_NETWORK_REQUIRED:
    return 3002
case .CANCELLED:
    return 3004
case .SESSION_UNSUCCESSFUL:
    return 3005
case .ENCRYPTION_KEY_INVALID:
    return 3006
case .TIMEOUT:
    return 3007
case .CONTEXT_SWITCH:
```

2. SDK Operations

```
        return 3008
    case .CAMERA_INITIALIZATION_ISSUE:
        return 3009
    case .UNKNOWN_INTERNAL_ERROR:
        return 3010
    case .LANDSCAPE_MODE_NOT_ALLOWED:
        return 3011
    case .REVERSE_PORTRAIT_NOT_ALLOWED:
        return 3012
    case .LOCKED_OUT:
        return 3013
    case .MISSING_GUIDANCE_IMAGES:
        return 3014
    case .USER_CANCELLED_VIA_CLICKABLE_READY_SCREEN_SUBTEXT:
        return 3015
    case .FACE_SCAN_LOW_MEMORY:
        return 3016
    case .GRACE_PERIOD_EXCEEDED:
        return 3017
    case .CAMERA_PERMISSION_ERROR:
        return 3019
```

2.8 Localizations NtdrSDK Strings resource

```
api?.sdkTheme?.selectIDTitleText = "Select ID type"
api?.sdkTheme?.selectIDDescriptionText = "Your ID document should be machine readable and should have a chip. If not select others "
api?.sdkTheme?.selectIDDescriptionText = "Your custom text"
api?.sdkTheme?.selectIDPassportOptionText = "Passport"
api?.sdkTheme?.selectIDResidentCardOptionText = "ID card"
api?.sdkTheme?.selectIDOtherCardOptionText = "Others"
```

"Facescan_Successmessage" = "Complete"; "Facescan_Uploadmessage" = "Still Uploading...";

"Select_Id_Type" = "Choose whether you want to use passport or ID card";

"Document_Scan_Info" = "Check if your passport or international ID card can be used on [MitlD.dk/idkrav](https://mitl.dk/idkrav)";

"Next_Button_Title_Doc_Option_Screen" = "Next";

"Retry_Limit_Title" = "Sorry";

"Retry_Limit_Message" = "Retry Limit To Connect To The Server Is Reached.";

Calling Apps Can Place Readid Strings In Separate 'Strings' File And Can Specify The Table Name In The Theme Customization Property Like Below:

Themecustomization.Passportscanviewcustomization.Customstringtablename = "Readid_Ui" //This Way You Can Separate The Strings From Your App's Strings In A Different File.

NtdrSdk Strings Resource For Document Flow

```
// Popup buttons
/**
Popups: readid_remove_document_message, readid_permission_required_error_title,
readid_wrong_document_selected_error_title
*/
"readid_ok" = "OK";
/**
Popups: readid_connection_error_title, readid_permission_required_error_title, readid_no_chip_warning_title
(on VIZ capture screen), readid_access_control_error_title, readid_nfc_not_enabled_title,
readid_not_supported_chip_warning_title, readid_wrong_document_selected_error_title
*/
"readid_cancel" = "Cancel";
// Popups: readid_permission_required_error_title, readid_nfc_not_enabled_title
"readid_settings" = "Settings";
// Popups: readid_access_control_error_title, readid_no_chip_warning_title
"readid_repeat_scan" = "Repeat scan";
/**
Popups: (only if skip reading is enabled) readid_access_control_error_title,
readid_not_supported_chip_warning_title, readid_no_chip_warning_title, readid_skip_reading (shown as a title)
*/
"readid_skip_reading" = "Skip reading";
// Popups: readid_not_supported_chip_warning_title, readid_no_chip_warning_title
"readid_try_anyway" = "Try anyway";
// Warnings and errors
// Connection error popup
"readid_connection_error_title" = "Connection error";
```


2. SDK Operations

```
"readid_connection_error_message" = "An error occurred while connecting to the server.
Check your internet connection, or try again later.";
"readid_try_again" = "Try again";
// Service unavailable popup
"readid_service_unavailable_error_title" = "Service unavailable";
"readid_service_unavailable_error_message" = "The service is temporarily not available.\nPlease try again later.";
// Server error popup
"readid_server_error_title" = "Server error";
"readid_server_error_message" = "An error has occurred while processing your request.\nPlease try again later.";
// Authorization error popup
"readid_authorization_error_title" = "Authorization error";
"readid_authorization_error_message" = "Authorization failed.\nPlease contact the support team.";
// Security error popup
"readid_connection_security_error_message" = "An error occurred while initializing a secure connection to the server.";
// Network error popup
"readid_unknown_error_title" = "Unexpected situation";
"readid_unknown_error_message" = "An unexpected situation occurred. Please try again later.";
// Only in MRZ SDK: Session timed out error popup
"readid_failed_reason_session_timeout" = "The ReadID session timed out";
// Session error popup
"readid_session_error" = "Cannot start session. Please try again later.";
// Network error on committing session popup
"readid_session_commit_error" = "Cannot send session. Please try again later.";
// Only in MRZ SDK: Session timed out error popup
"readid_session_timeout" = "The session timed out. Please start again.";
// Only in MRZ SDK: Missing face image popup
"readid_fatal_error_missing_face_image" = "A valid face image could not be read from the document.";
// System popup
"readid_please_wait" = "Please wait...";
// Popup title: readid_connection_security_error_message, readid_session_error, readid_session_commit_error
"readid_error" = "Error";
// Only in modal view: Confirmation popup to dismiss the modal
"readid_confirm" = "Confirm";
"readid_cancel_message" = "Are you sure you want to stop?";
"readid_close" = "Close";
// Top navigation bar: back button
"readid_back_navigation_title" = "";
// Document selection screen
// Title
"readid_document_selection_title" = "Select document type";
// Description
"readid_selection_text" = "What type of identity document do you want to use?";
// Buttons
"readid_passport" = "Passport";
"readid_identity_card" = "ID card";
"readid_id_card" = "ID card";
"readid_french_identity_card" = "French identity card";
"readid_driving_licence" = "Driver's license";
"readid_drivers_license" = "Driver's license";
"readid_vehicle_registration" = "Vehicle registration";
"readid_visa" = "Visa";
// VIZ animation screen
// Title
"readid_scan_document" = "Scan the document";
// Title in the top navigation bar (only shown when the logo is disabled)
"readid_mrz_animation_title_passport" = "";
"readid_mrz_animation_title_identity_card" = "";
"readid_mrz_animation_title_driving_licence" = "";
"readid_mrz_animation_title_id_card" = "How to scan ID card";
"readid_mrz_animation_title_drivers_license" = "How to scan driver's license";
"readid_mrz_animation_title_visa" = "How to scan visa";
"readid_mrz_animation_title_vehicle_registration" = "How to scan vehicle registration";
// Descriptions
"readid_mrz_passport_explanation" = "Open to the photo page";
"readid_mrz_id_card_explanation" = "Rotate to the back";
"readid_mrz_drivers_license_explanation" = "Keep the front facing up";
"readid_mrz_visa_explanation" = "Scan the MRZ on your visa";
"readid_mrz_vehicle_registration_explanation" = "Scan the MRZ on your document";
"readid_mrz_explanation" = "Adjust until the text turns green";
"readid_mrz_identity_card_front_explanation" = "Keep the front facing up";
"readid_mrz_identity_card_back_explanation" = "Flip the identity card to the back";
"readid_mrz_driving_licence_front_explanation" = "Keep the front of the driving licence facing up";
"readid_mrz_driving_licence_back_explanation" = "Rotate to the back";
"readid_mrz_passport_data_page_front_explanation" = "Open the passport to the photo page";
"readid_mrz_passport_data_page_back_explanation" = "Open back of the photo page";
// Button
"readid_viz_animation_primary_button" = "Next";
// Camera permission error popup
"readid_permission_required_error_title" = "Permission required";
// Only shown if camera permission needs to be enabled in the app settings
"readid_grant_permission_in_app_settings" = "Please allow camera access in the application settings.";
// VIZ capture screen
// Titles
"readid_scan_viz_identity_card_front" = "Align front of identity card here";
"readid_scan_viz_identity_card_back" = "Align back of identity card here";
```

2. SDK Operations

```
"readid_scan_viz_driving_licence_front" = "Align front of driving license here";
"readid_scan_viz_driving_licence_back" = "Align back of driving license here";
"readid_scan_viz_passport_front" = "Align passport photo page here";
"readid_scan_viz_passport_back" = "Align back of passport photo page here";
"readid_scan_viz_visa_front" = "Align visa page here";
"readid_scan_mrz" = "Scan MRZ";
"readid_accessing_nfc_chip_hold_still" = "Please hold still";
// Alternates with the title every few seconds
"readid_avoid_white_background" = "Avoid white background";
// Title in the top navigation bar
"readid_mrz_scan_title_passport" = "Scan passport";
"readid_mrz_scan_title_id_card" = "Scan id card";
"readid_mrz_scan_title_drivers_license" = "Scan driver's license";
"readid_mrz_scan_title_identity_card" = "Scan identity card";
"readid_mrz_scan_title_driving_licence" = "Scan driving licence";
"readid_mrz_scan_title_visa" = "Scan visa";
"readid_mrz_scan_title_vehicle_registration" = "Scan vehicle registration";
"readid_mrz_result_title_passport" = "Passport data";
"readid_mrz_result_title_id_card" = "ID card data";
"readid_mrz_result_title_drivers_license" = "Driver's license data";
"readid_mrz_result_title_visa" = "Visa data";
"readid_mrz_result_title_vehicle_registration" = "Vehicle registration data";

// Dynamic feedback messages
"readid_hold_document_closer" = "Hold document closer to camera";
"readid_no_finger_on_document" = "Do not cover document with fingers";
"readid_no_glare_on_document" = "Reduce glare on document";
"readid_document_out_of_focus" = "Document out of focus";
"readid_no_mrz_detected" = "No MRZ detected";
"readid_no_face_image_detected" = "No face image detected";
"readid_no_qr_code_detected" = "No QR code detected";
"readid_viz_scan_in_progress" = "Scanning in progress";
"readid_document_scanned" = "Document scanned";
// Manual input button
"readid_manual_input" = "Manual input";
"readid_manual_input_button_description" = "Manual input";
"readid_manual_input_button_accessibility_hint" = "Open screen for manual input.";
// Torch button
"readid_torch_on" = "Torch";
"readid_torch_off" = "Torch";
"readid_torch_button_accessibility_label" = "Torch";
"readid_torch_button_accessibility_hint" = "Turn the torch on or off.";
"readid_torch_on_accessibility_value" = "On";
"readid_torch_off_accessibility_value" = "Off";
// Manual capture button (only possible with VIZOnlyOnePageFlow)
"readid_manual_capture_button_accessibility_hint" = "Manually take a picture.";
// No MRZ error popup
"readid_no_mrz_warning_message" = "The document you scanned does not contain a Machine Readable Zone or the scan did not work.";
// No NFC chip error popup on VIZ capture
"readid_no_chip_error_message" = "The document you scanned does not contain an NFC chip or the scan did not work.";
"readid_please_repeat_scan" = "Please repeat scan.";
"readid_repeat_or_continue" = "Please repeat scan or continue with the second page.";
"readid_continue" = "Continue";
// Back button
"readid_viz_scan_back_button_accessibility_label" = "Back";
// Manual input screen
// Title
"readid_title_input" = "Enter document data";
// Segmented buttons (+ readid_document_number for driving licences instead of readid_mrz)
"readid_mrz" = "MRZ";
"readid_can" = "CAN";
// MRZ page
// X button
"readid_x_button_accessibility_hint" = "In case the date of birth contains an X. Use this button to add it to the field.";
// ICAO input labels (+ readid_document_number)
// %1$s will be replaced with the date format
"readid_date_of_birth_input" = "Date of birth (%@)";
// %1$s will be replaced with the date format
"readid_date_of_expiry_input" = "Date of expiry (%@)";
// Driving license input label
"readid_mrz_input" = "Machine Readable Zone (MRZ)";
// CAN page
// Input labels (for driving licences we use readid_document_number)
"readid_can_input" = "Card Access Number (CAN)";
// ICAO description
"readid_can_can_input_description" = "Using this input to access the document chip is supported by a limited number of documents. Make sure that you selected the right document type and that your document supports PACE-CAN validation before continuing.

Known unsupported documents:
\U2022 Ukrainian ID-Card";
// Driving license description
"readid_can_document_number_input_description" = "Using this input to access the document chip is supported by second generation Dutch driving licence. Make sure that your Machine Readable Zone (MRZ) starts with \"D1NLD2\".";
// Button
```

2. SDK Operations

```
"readid_viz_manual_input_primary_button" = "Next";
"readid_manual_input_keyboard_button" = "Done";
// Input errors
"readid_manual_input_required" = "Required";
// %1$s will be replaced with the label of the manual input field
"readid_manual_input_required_error" = "Required";
// %1$s will be replaced with the label of the manual input field
"readid_manual_input_invalid_format_error" = "Invalid format";
// %1$s will be replaced with the label of the manual input field, %2$s will be replaced with the date format
"readid_manual_input_invalid_date_error" = "Please enter a valid date format.";
// VIZ confirmation screen
// Description
"readid_viz_confirmation" = "Make sure the document is fully visible and does not show any glare or fingers. If needed, repeat the scan.";
// Buttons
"readid_viz_confirmation_primary_button" = "Next";
"readid_viz_confirmation_secondary_button" = "Repeat scan";
// VIZ result screen
// Title
"readid_viz_result_title" = "Document scanned";
// Segmented buttons (+ readid_data in VIZ and NFC result screen section)
"readid_photos" = "Photos";
"readid_performance" = "Performance";
// Button
"readid_viz_result_primary_button" = "Next";
// Photo page
// Section headers (also used on VIZ confirmation screen)
"readid_front_of_document" = "Front of document";
"readid_back_of_document" = "Back of document";
"readid_photo_page" = "Photo page";
"readid_back_of_photo_page" = "Back of photo page";
// Section headers (also used on VIZ confirmation screen) and label labels on "Performance" page
"readid_identity_card_front" = "ID card front";
"readid_identity_card_back" = "ID card back";
"readid_driving_licence_front" = "Driving licence front";
"readid_driving_licence_back" = "Driving licence back";
// Accessibility labels (also used on VIZ confirmation screen)
"readid_viz_image_description" = "VIZ";
"readid_qr_code_image_description" = "QR code";
// Data page
// Section header
"readid_collected_information" = "Collected information";
"readid_identification_number" = "Identification number";
"readid_brand" = "Brand";
"readid_model" = "Model";
"readid_category" = "Category";
"readid_classification" = "Classification";
"readid_registration_number" = "Registration number";
"readid_date_of_first_registration" = "Date of first registration";
"readid_mrz_text" = "Recognized MRZ";
"readid_mrz_image" = "MRZ image";
"readid_viz_image" = "VIZ image";
"readid_qr_code_result" = "QR code result";
// Table labels
"readid_total_ocr_time" = "Total OCR time";
"readid_net_ocr_time" = "Net OCR time";
"readid_vehicle_information" = "Vehicle information";
"readid_ocr_information" = "OCR information";
"readid_issuing_department" = "Issuing department";
"readid_issuing_office" = "Issuing office";
"readid_value" = "Value";
// Accessibility labels
"readid_mrz_image_description" = "MRZ";
// Performance page
// Table labels
"readid_capture_mode" = "Capture mode";
"readid_page_type" = "Page type";
"readid_ensure_sharpness" = "Ensure sharpness";
"readid_prevent_glare_on_document" = "Prevent glare on document";
"readid_prevent_finger_on_document" = "Prevent finger on document";
"readid_mrz_result" = "MRZ result";
"readid_face_image_result" = "Face image result";
// Screen values
"readid_passed" = "Passed";
"readid_failed" = "Failed";
"readid_degraded" = "Degraded";
"readid_disabled" = "Disabled";
"readid_automatic" = "Automatic";
"readid_automatic_degraded" = "Automatic and degraded";
"readid_manual" = "Manual";
// VIZ and NFC result screen
// Segmented button
"readid_data" = "Data";
// Section headers
"readid_document_information" = "Document information";
"readid_personal_information" = "Personal information";
```

2. SDK Operations

```
"readid_optional_information" = "Optional information";
// Table labels
"readid_last_name" = "Name";
"readid_first_name" = "Given names";
"readid_gender" = "Gender";
"readid_document_code" = "Document code";
"readid_issuing_country" = "Issuing country";
"readid_nationality" = "Nationality";
"readid_document_number" = "Document number";
"readid_date_of_birth" = "Date of birth";
"readid_date_of_expiry" = "Date of expiry";
"readid_optional_data" = "Optional data";
"readid_optional_data_2" = "Optional data 2";
"readid_personal_number" = "Personal number";
"readid_date_of_issue" = "Date of issue";
// Accessibility labels (also used on VIZ confirmation screen)
"readid_maximise_image" = "Maximise";
"readid_close_image" = "Close";
// NFC animation screen
// Title
"readid_read_chip" = "Read the NFC chip";
// Description
"readid_touch_to_read" = "Place the phone on top of the document.";
"readid_hold_together" = "Hold together and keep still";
// NFC instruction carousel
// Accessibility labels
"readid_next_instruction_button_description" = "Next instruction";
"readid_previous_instruction_button_description" = "Previous instruction";
// Passport descriptions
"readid_carousel_general_remove_document_covers" = "Remove the cover";
"readid_carousel_general_check_symbol" = "Check for the symbol";
"readid_carousel_passport_keep" = "Keep the passport and the phone close to each other";
"readid_carousel_passport_photo_page" = "Try reading the photo page";
"readid_carousel_passport_last_page" = "Try reading the inner back page";
// Only in MRZ SDK
"readid_animation_nfc_open_passport" = "Open the passport to the back page";
// Identity card description
"readid_carousel_identity_card_keep" = "Keep the identity card and the phone close to each other";
// Driving license description
"readid_carousel_driving_licence_keep" = "Keep the driving license and the phone close to each other";
"readid_carousel_id_card_keep" = "Keep the ID card and the phone close to each other";
"readid_carousel_drivers_license_keep" = "Keep the driver's license and the phone close to each other";
"readid_carousel_general_keep_still" = "Keep still during reading";
// General description
"readid_carousel_general_slide_up_down" = "Slide the phone up and down slowly until the progress bar appears";
// NFC read screen
// Title
"readid_nfc_read_title" = "Read the NFC chip";
// Title in the top navigation bar (only shown when the logo is disabled)
"readid_nfc_scan_title_passport" = "Read passport chip";
"readid_nfc_scan_title_id_card" = "Read ID card chip";
"readid_nfc_scan_title_drivers_license" = "Read driver's license chip";

"readid_nfc_scan_title_identity_card" = "";
"readid_nfc_scan_title_driving_licence" = "Read driving licence chip";
"readid_nfc_scan_title_visa" = "Read visa chip";
"readid_nfc_scan_title_vehicle_registration" = "Read vehicle registration chip";
```

2. SDK Operations

```
// NFC toaster/sheet
"readid_waiting_for_chip" = "Hold the middle of the document to the top of the phone";
"readid_waiting_for_chip_passport" = "Hold the middle of the passport to the top of the phone";
"readid_waiting_for_chip_id_card" = "Hold the middle of the ID card to the top of the phone";
"readid_waiting_for_chip_drivers_license" = "Hold the middle of the driver's license to the top of the phone";
// Buttons
"readid_nfc_read_secondary_button" = "Skip reading";
"readid_nfc_read_primary_button" = "Next";
// Progress bar
"readid_accessing_nfc_chip" = "Accessing NFC chip";
"readid_reading_personal_data" = "Reading personal data";
"readid_reading_photo" = "Retrieving photo...";
"readid_checking_validity" = "Checking validity";
"readid_chip_read" = "NFC chip reading completed";
// Access control error popup
"readid_access_control_error_title" = "Authentication failed";
"readid_access_control_error_message" = "The access control information is probably incorrect.\nPlease validate the collected information and try again.";
"readid_access_control_error_without_manual_input_message" = "The access control information is probably incorrect. Please scan the MRZ again.";
// Lost NFC connection message
"readid_starting_over" = "Lost connection\nTry not to move the phone...";
// NFC resume message
"readid_nfc_resume" = "Reading interrupted.
Make sure the phone is on top of the document.";
// Skip reading popup
"readid_confirm_skip_message" = "Do you want to skip reading the document? The identification process might not succeed if you choose to skip.";
"readid_skip" = "Skip";
"readid_check_input" = "Check input";
"readid_keep_trying" = "Keep trying";
// No NFC chip error popup
"readid_no_chip_warning_title" = "No NFC chip";
"readid_no_chip_warning_message" = "The document you are trying to read most likely does not have an NFC chip. If you continue, the reading may fail.";
// Not supported chip error popup
"readid_not_supported_chip_warning_title" = "Not supported NFC chip";
"readid_not_supported_chip_warning_message" = "The document you are trying to read is most likely not supported by us. If you continue, the reading may fail.";
// Wrong document selected error popup
"readid_wrong_document_selected_error_title" = "Wrong document selected";
"readid_wrong_document_selected_error_message" = "Please go back and select the right document type before trying again.";
// Only in MRZ SDK: unsupported error popup
"readid_access_control_unsupported_error_title" = "Document reading failed";
"readid_access_control_unsupported_error_message" = "This document is unsupported. Try again with a valid document.";
// Session commit popup
"readid_sending_data" = "Sending data...";
// NFC result screen
// Titles
"readid_nfc_result_title_passport" = "Passport chip data";
"readid_nfc_result_title_id_card" = "ID card chip data";
"readid_nfc_result_title_drivers_license" = "Driver's license chip data";
"readid_nfc_result_title_visa" = "Visa chip data";
"readid_nfc_result_title_vehicle_registration" = "Vehicle registration chip data";

"readid_nfc_result_title_identity_card" = "Identity card chip data";
"readid_nfc_result_title_driving_licence" = "Driving licence chip data";
// Segmented buttons (+ readid_data in VIZ and NFC result screen section)
"readid_security" = "Security";
// No content message when server echo is disabled or filtered
"readid_no_document_information" = "No document information";
"readid_no_document_information_message" = "You don't have permission to see any information from the document read.";
// Button
"readid_nfc_result_primary_button" = "Next";
// Data page
// Section headers
"readid_face_image" = "Face image";
"readid_verification_information" = "Validity";
"readid_signature" = "Signature";
"readid_mr_z_from_chip" = "MRZ from chip";
"readid_sai_from_chip" = "SAI from chip";
// Table labels
"readid_face_image_description" = "Face";
"readid_verification_content_authenticity" = "Content authenticity";
"readid_verification_chip_authenticity" = "Chip authenticity";
"readid_verification_expiration_status" = "Expiration status";
"readid_verification_result" = "Verification result";
"readid_title" = "Title";
"readid_name_of_holder" = "Full name";
"readid_other_names" = "Other names";
"readid_place_of_birth" = "Place of birth";
"readid_country_of_birth" = "Country of birth";
"readid_address" = "Address";
"readid_profession" = "Profession";
"readid_telephone" = "Telephone";
"readid_custodian" = "Custodian";
```

2. SDK Operations

```
"readid_issuing_authority" = "Issuing authority";
"readid_drivers_licenses" = "Driver's licenses";
"readid_driving_licences" = "Driving licences";
"readid_driving_licence_category" = "Category";
// Screen values
"readid_verification_succeeded" = "Authentic content and chip";
"readid_verification_not_supported" = "Data authentic, clone detection not supported by this document";
"readid_verification_passed" = "Data authentic, clone detection not performed";
"readid_verification_failed" = "Authenticity not validated";
"readid_verification_expired" = "Document expired";
"readid_verification_content_authenticity_succeeded" = "Authentic content";
"readid_verification_content_authenticity_failed" = "Authenticity not validated";
"readid_verification_chip_authenticity_succeeded" = "Authentic chip";
"readid_verification_chip_authenticity_failed" = "Authenticity not validated";
"readid_verification_chip_authenticity_not_present" = "Clone detection not supported by this document";
"readid_verification_chip_authenticity_not_checked" = "Clone detection not performed";
"readid_verification_expiration_status_valid" = "Document not expired";
"readid_verification_expiration_status_expired" = "Document expired";
"readid_value_gender_male" = "Male";
"readid_value_gender_female" = "Female";
"readid_value_unspecified" = "Unspecified";
"readid_value_unknown" = "Unknown";
// Security page
// Section headers
"readid_chip_information" = "Chip information";
"readid_validity_information" = "Validity information";
"readid_document_signing_certificate" = "Document signing certificate";
"readid_country_signing_certificate" = "Country signing certificate";
// Table labels
// Chip information
"readid_chip_ids_version" = "LDS version";
"readid_chip_data_groups" = "Data groups";
// Validity information
"readid_type_access_control" = "Type of access control";
"readid_active_authentication" = "Active authentication";
"readid_chip_authentication" = "Chip authentication";
"readid_data_group_hashes" = "Data group hashes";
"readid_document_signer" = "Document signer";
"readid_country_signer" = "Country signer";
// Certificate information
"readid_link_certificate" = "Link certificate";
"readid_cert_serial_number" = "Serial number";
"readid_cert_signature_alg" = "Signature algorithm";
"readid_cert_pubkey_alg" = "Public key algorithm";
"readid_cert_thumbprint" = "Certificate thumbprint";
"readid_cert_issuer" = "Issuer";
"readid_cert_valid_from" = "Valid from";
"readid_cert_valid_to" = "Valid to";
"readid_cert_subject" = "Subject";
// Data group hashes information
"readid_data_group_hashes_information" = "Data group hashes information";
// %1$s will be replaced with the data group number
"readid_data_group_hash" = "Data group %1$d";
"readid_stored_hash" = "• stored";
"readid_computed_hash" = "• computed";
// Screen values
"readid_verification_verdict_success" = "SUCCEEDED";
"readid_verification_verdict_failed" = "FAILED";
"readid_verification_verdict_not_checked" = "NOT CHECKED";
"readid_verification_verdict_not_present" = "NOT PRESENT";
"readid_verification_verdict_unknown" = "UNKNOWN";
"readid_verification_reason_unknown" = "Unknown";
"readid_verification_reason_succeeded" = "Succeeded";
"readid_verification_reason_not_supported" = "Not supported";
"readid_verification_reason_unexpected_exception_failure" = "Unexpected exception";
"readid_verification_reason_unsupported_key_type_failure" = "Unsupported key type";
"readid_verification_reason_unsupported_signature_algorithm_failure" = "Unsupported signature algorithm";
"readid_verification_reason_unsupported_digest_algorithm_failure" = "Unsupported digest algorithm";
"readid_verification_reason_signature_checked" = "Signature checked";
"readid_verification_reason_signature_failure" = "Signature failure";
"readid_verification_reason_read_error_sod_failure" = "Error reading security object";
"readid_verification_reason_found_a_chain_succeeded" = "Found a chain to a trust anchor";
"readid_verification_reason_could_not_build_chain_failure" = "Could not build a chain to a trust anchor";
"readid_verification_reason_no_trust_anchors_found_failure" = "No trust anchors found";
"readid_verification_reason_all_hashes_match" = "All hashes match";
"readid_verification_reason_stored_hash_not_found_failure" = "Stored hash not found";
"readid_verification_reason_hash_mismatch_failure" = "Found a mismatching hash";
"readid_verification_reason_read_error_configuration" = "Error reading configuration";
"readid_verification_reason_certificate_expired" = "Certificate expired";
"readid_verification_reason_presence_detected" = "Presence detected";
"readid_verification_reason_non_match_alert" = "Non match alert";
"readid_not_available" = "NOT AVAILABLE";
"readid_hash_match" = "MATCH";
"readid_hash_mismatch" = "MISMATCH";
"readid_hash_not_computed" = "NOT COMPUTED";
```


2. SDK Operations

```
"FaceTec_feedback_face_not_looking_straight_ahead" = "Look Straight Ahead";
"FaceTec_feedback_face_not_upright" = "Hold Your Head Straight";
"FaceTec_feedback_hold_steady" = "Hold Steady";
"FaceTec_feedback_move_phone_away" = "Move Away";
"FaceTec_feedback_move_phone_closer" = "Move Closer";
"FaceTec_feedback_move_phone_even_closer" = "Even Closer";
"FaceTec_feedback_move_phone_to_eye_level" = "Move Camera To Eye Level";
"FaceTec_feedback_use_even_lighting" = "Light Face More Evenly";

"FaceTec_idscan_type_selection_header" = "Prepare to scan\nYour ID document";
"FaceTec_idscan_capture_tap_to_focus_message" = "Tap screen to focus";
"FaceTec_idscan_capture_hold_steady_message" = "Please hold steady";
"FaceTec_idscan_capture_id_front_instruction_message" = "Show front of ID";
"FaceTec_idscan_capture_id_back_instruction_message" = "Show back of ID";
"FaceTec_idscan_review_id_front_instruction_message" = "Confirm photo is clear & legible";
"FaceTec_idscan_review_id_back_instruction_message" = "Confirm photo is clear & legible";
"FaceTec_idscan_ocr_confirmation_main_header" = "Review & confirm";
"FaceTec_idscan_ocr_confirmation_scroll_message" = "Scroll down";

"FaceTec_instructions_header_ready_1" = "Please place your face";
"FaceTec_instructions_header_ready_2" = "in the frame";
"FaceTec_instructions_message_ready_1" = " ";
// "FaceTec_instructions_message_ready_2" = "Press I'm Ready & Move Closer";
"FaceTec_instructions_message_ready_2" = " ";

"FaceTec_presession_frame_your_face" = "Frame Your Face In The Oval";
"FaceTec_presession_position_face_straight_in_oval" = "Look Straight Ahead";
"FaceTec_presession_hold_steady_3" = "Hold Steady For: 3";
"FaceTec_presession_hold_steady_2" = "Hold Steady For: 2";
"FaceTec_presession_hold_steady_1" = "Hold Steady For: 1";
"FaceTec_presession_eyes_straight_ahead" = "Look Straight Ahead";
"FaceTec_presession_remove_dark_glasses" = "Remove Dark Glasses";
"FaceTec_presession_neutral_expression" = "Neutral Expression, No Smiling";
"FaceTec_presession_conditions_too_bright" = "Conditions Too Bright";
"FaceTec_presession_brighten_your_environment" = "Brighten Your Environment";

"FaceTec_result_facescan_upload_message" = "Processing...";
"FaceTec_result_success_message" = "Completed";
// "FaceTec_result_idscan_upload_message" = "Uploading\nEncrypted\nID Scan";
"FaceTec_result_idscan_upload_message" = "Processing...";
"FaceTec_result_idscan_unsuccess_message" = "ID Photo\ndid not match\nuser's face";
"FaceTec_result_idscan_success_front_side_message" = "Your 3D face\nmatched Your ID";
"FaceTec_result_idscan_success_front_side_back_next_message" = "ID scan complete";
```


2. SDK Operations

```
"FaceTec_result_idscan_success_back_side_message" = "ID scanned\nsuccessfully";  
"FaceTec_result_idscan_success_user_confirmation_message" = "Photo ID scan\nincomplete";  
"FaceTec_result_idscan_retry_face_did_not_match_message" = "Face did not match highly\n\nPlease try again";  
"FaceTec_result_idscan_retry_id_not_fully_visible_message" = "ID document\n\nnot fully visible";  
"FaceTec_result_idscan_retry_ocr_results_not_good_enough_message" = "ID text not legible\n\nPlease try again";  
"FaceTec_result_idscan_retry_id_type_not_supported_message" = "ID type not supported\n\nPlease use a different ID";  
"FaceTec_result_idscan_retry_barcode_not_read_message" = "Barcode failed to scan\n\nPlease try again";  
  
"FaceTec_retry_header" = "Let's try that again...";  
"FaceTec_retry_subheader_message" = "We need a clearer video selfie.";  
"FaceTec_retry_instruction_message_1" = "Please ensure that light does not create shadows or glare\n";  
"FaceTec_retry_instruction_message_2" = "Ensure video is not shaky";  
"FaceTec_retry_instruction_message_3" = "Neutral Expression, No Smiling";  
"FaceTec_retry_your_image_label" = "Your Selfie";  
"FaceTec_retry_ideal_image_label" = "Ideal Pose";
```