Vantage Mobile Input

Developer's Guide

Table of Contents

Vantage Mobile Input

Vantage Mobile Input can power your application with the following features:

- **Out-of-the-box mobile image capture**: Enables capturing documents with a mobile device's camera, using UI components to obtain the best quality images suitable for OCR and for further processing.
- **Automatic document detection**: Detects document edges, crops images, and corrects perspective.
- **Customizable document structure**: Guides users through the capturing process according to predefined types and numbers of documents.
- **Ability to work in connected and disconnected modes**: Allows instantly capturing and uploading documents when Internet access is available, or capturing and saving for later upload.
- **Access to ABBYY Vantage**: Allows users to process document images using ABBYY Vantage skills.

Vantage Mobile Input can be integrated into native Android and iOS applications. For more information, see Integrating Vantage Mobile Input to native applications.

Vantage Mobile Input can also serve as a ready-to-use micro application for iOS and Android or as a web interface. On using Vantage Mobile Input with the Vantage API, see Vantage Mobile Input with the Vantage API.

Vantage Mobile Input with the Vantage API

You can use the Vantage Mobile Input feature, which allows capturing, processing, and uploading documents directly from the mobile device to ABBYY Vantage.

Vantage Mobile Input starts with obtaining a mobile upload link via the Vantage API. The link will contain the parameters defining the number and types of documents to be captured during the mobile input session, ABBYY Vantage skills to process uploaded documents and more. For more information about using the Vantage API, see the Swagger documentation.

To add the Vantage Mobile Input feature by getting the mobile upload link for the transaction, complete the following steps:

1. Authenticate in ABBYY Vantage (see Authentication for detailed instructions).

Note: Each HTTP request to ABBYY Vantage should have a header containing an access token which the service will use to identify user in the system.

2. View the list of available skills and find the ID of the skill you need to process documents.
   To do so, send a **GET** request to the **skills** resource:

```
GET https://vantage.abbyy.com/api/publicapi/v1/skills
```

Run the following command:

▾ For Windows

```
curl -X GET "https://vantage.abbyy.com/api/publicapi/v1/skills" \

-H "Authorization: Bearer token"
```

For Linux

```
curl -X GET 'https://vantage.abbyy.com/api/publicapi/v1/skills' \

-H 'Authorization: Bearer token'
```

The response will contain a JSON file with the list of available skills. Find the skill identifier and use it as the value for the **skillId** parameter to create a transaction.

Response

```
[
   {
      "id": "c4b26798-07cb-11eb-adc1-0242ac120002",
      "name": "NewClassifier",
      "type": "Classification"
   }
   {
      "id": "c4b26798-07cb-11eb-adc1-0242ac120002",
      "name": "Invoice",
      "type": "Document"
   }
]
```

3. Create a transaction for the selected skill and get a mobile upload link.
   To create transaction, send a **POST** request to the **Transactions** resource:

```
POST https://vantage.abbyy.com/api/publicapi/v1/transactions
```

Run the following command:

For Windows

```
curl -X POST "https://vantage.abbyy.com/api/publicapi/v1/transactions" \

-H "Authorization: Bearer token" \
-H "Content-Type: application/json" \
-d "{\"skillId\":\"123\", \"generateMobileInputLink\": true}"
```

For Linux

```
curl -X POST 'https://vantage.abbyy.com/api/publicapi/v1/transactions' \

-H 'Authorization: Bearer token' \
-H 'Content-Type: application/json' \
-d '{\"skillId\":\"123\", \"generateMobileInputLink\": true}'
```

After the transaction has been successfully created, you will receive a response with the transaction identifier and the mobile upload link:

Response

```
{
   "transactionId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
   "mobileInputLink": "https://vantage.abbyy.com/mobile?baseUrl=<base
URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>"
}
```

The link will be valid for 30 minutes, after which a new link should be obtained.

Note: If you use QR codes or buttons for uploading documents on your website, we recommend that these QR codes and buttons create a new transaction and/or refresh the token every 30 minutes. Otherwise, users will not be able to upload documents after the initial token expires.

Using the obtained link, you can start Vantage Mobile Input with the Vantage API, configure with additional parameters, track the processing, and download the result. Instructions on how to take photos of documents and upload documents from a mobile device with the micro application for iOS and Android are available in the Uploading Documents from a Mobile Device section of the Quick Start Guide.

Configuring Vantage Mobile Input

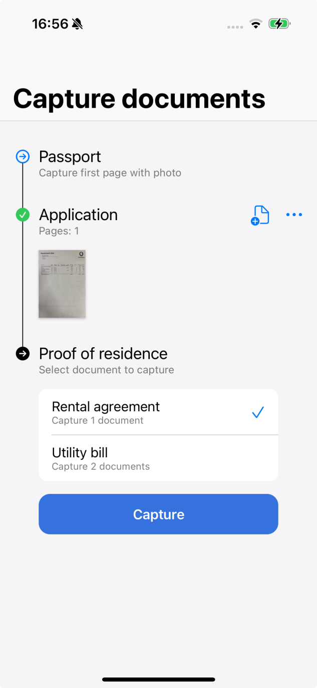
You can configure the mobile upload feature using the following additional parameters:

- The md parameter customizes the number and names of the documents, along with the pages for each document to be captured in one transaction.
- The dt parameter customizes mobile upload to capture a single document type.
- The redirect_uri parameter customizes a website link that will be opened once the mobile upload is finished.
- The ma parameter enables the use of micro applications for on-premises environments.

The md parameter

You can specify the structure of the documents with the number, types, and names of documents that need to be processed in one transaction using the **md** parameter. Vantage Mobile Input will guide users on how to accurately capture specified documents for further processing.

▾ Screenshot demonstrating the document structure configured using the md parameter in the mobile upload



16:56

Capture documents

Passport
Capture first page with photo

Application
Pages: 1

Proof of residence
Select document to capture

Rental agreement ✓
Capture 1 document

Utility bill
Capture 2 documents

Capture

By default, the **md** parameter is present in the mobile upload link with a value of either 0 or 1, depending on the selected skill. You can specify one of the following values:

Value	Description
0	Captures one document with any number of pages.
1	Captures any number of documents with any number of pages.
Encoded URL to JSON file	Captures a specified number of documents with customized names, along with defined number and names of pages for each document.
Encoded JSON file	

Example of using the encoded URL to JSON file for the **md** parameter:

```
https://vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&md=https%3A%2F%2Fdomain.tld%2Fstructure.json&token=<token>&v=<2.4>
```

Example of using the encoded JSON file for the **md** parameter:

```
https://vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&md=<encoded_json_structure>&token=<token>&v=<2.4>
```

Note: If the dt parameter is used in the mobile upload link, the value of the **md** parameter will be ignored.

The dt parameter

To customize mobile upload to capture a single document type, use the **dt** parameter:

1. Add the dt parameter to the mobile upload link.
2. Specify one of the following values:

Value	Description
idcard	Captures both sides (front and back) of an ID card.
passport	Captures the first page of a passport.
document	Captures any other document.

Note: The **dt** parameter is optional and does not have a default value.

Using the **dt** parameter will adjust the camera in the micro application so that identity documents are captured automatically.

Example of using the **dt** parameter for capturing ID cards:

```
https://vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>&dt=idcard
```

Example of using the **dt** parameter for capturing a passport page:

```
https://vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>&dt=passport
```

⚠ **Important!** When the **dt** parameter is used in the mobile upload link, only one document can be uploaded in a single transaction regardless of the selected skill.

The redirect_uri parameter

To specify a URL to which the user will be redirected once the mobile upload is finished, use the **redirect_uri** parameter:

1. Add the **redirect_uri** parameter to the mobile upload link.
2. Specify your intended redirect URL as its value.

Example of using the **redirect_uri** parameter:

```
https://vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>&redirect_uri=https%3A%2F%2Fyourwebsite.com
```

The ma parameter

By default, mobile upload for on-premises environments is only possible using a web browser. To enable the use of micro applications for on-premises environments, use the **ma** parameter:

1. Add the **ma** parameter to the mobile upload link.
2. Specify one of the ABBYY Vantage cloud environments.

In this case, mobile input will be redirected to one of the ABBYY Vantage cloud environments to start a micro application on the user's mobile device. All documents will be uploaded directly to the on-premises installation of ABBYY Vantage.

Examples of using the **ma** parameter:

`https://v2dev2-dev.vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>`*`&ma=<https%3A%2F%2Fvantage-eu.abbyy.com>`* (located in Western Europe),

`https://v2dev2-dev.vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>`*`&ma=<https%3A%2F%2Fvantage-us.abbyy.com>`* (located in North America),

`https://v2dev2-dev.vantage.abbyy.com/mobile?baseUrl=<base URL>&transactionId=<transaction-id>&token=<token>&v=<2.4>`*`&ma=<https%3A%2F%2Fvantage-au.abbyy.com>`* (located in Australia).

JSON documents schema

JSON files contain data structures specifying the number, types, and names of the documents, as well as the number and names of the pages for each document. To configure mobile upload for a specific user's scenarios, use the JSON file in the mobile upload link.

The **md** parameter can accept a JSON file in two formats: direct encoded link or encoded JSON.

To use an encoded link:

1. Generate a direct access link to the JSON file.

Note: The link should start with "http://" or "https://".

2. Encode the obtained link to URL-encoded format.
3. Specify the encoded URL as the value for the **md** parameter in the mobile upload link.

To use an encoded JSON:

1. Encode the JSON file to URL-encoded format.
2. Specify the encoded JSON file as the value for the **md** parameter in the mobile upload link.

JSON structure

TransactionStructure defines a set of documents expected to be captured by the user during a mobile upload session for further processing in a transaction. The transaction will not start until all documents are captured.

Properties	Value	Description
version	2.4	The current ABBYY Vantage version. **Note:** The value should correspond to the **v** parameter value in the mobile upload link.
elements	[ElementStructure]	The structure of the collection of elements.

ElementStructure defines a collection of one or multiple sets of documents expected to be captured by the user for further processing in a transaction.

Properties	Value	Description
type	standalone	Defines a collection with one set of documents.
	single_choice	Defines a collection with multiple sets of documents. **Note:** Requires at least two specified sets of documents.
name		The name of the set of documents that will be displayed on the **Capture Documents** screen in the micro application. **Note:** Available only for the **single_choice** type.
document	[DocumentStructure]	The structure of one document in the set. **Note:** Available only for the **standalone** type.
documents	[DocumentStructure]	The structure of all documents in the set. **Note:** Available only for the **single_choice** type.

The **DocumentStructure** defines a collection of the properties for each document in the set.

Properties	Value	Description
type	id_card	Defines capturing both sides (front and back) of an ID card.
	passport	Defines capturing the first page of a passport.
	document	Defines capturing any documents.
name		The name of the document that will be displayed on the **Capture Documents** screen in the micro application.
count	1,2,3...	The required number of documents to be captured in one transaction. The default value is "1".

Properties	Value	Description
		Note: The upload will not start until all documents are captured.
	any	The number of documents to be captured in one transaction. **Note:** The upload will start with any number of captured documents.
galleryEnabled	false/true	Defines whether uploading images of document pages from the device Gallery is allowed for this document.
pages	[PagesStructure]	The structure of the document pages.

The **PagesStructure** defines a collection of properties for the document pages expected to be captured by the user.

Properties	**Value**	**Description**
name	Front side/Back side	The names of the pages of an ID card.
	First page with photo	The name of the first page of a passport.
	Document page name	The names of the pages of any document.
count	1,2,3...	The required number of pages to be captured for the document. The default value is "1". The upload will not start until all pages of the document are captured. **Note:** The required value for the front and back side of an ID card is "1".
	any	The number of pages to be captured for any document. The upload will start with any number of captured pages.

		Note: The required value for the first page of a passport is "1". The required value for other passport pages is "any".

JSON sample

This JSON sample defines the document structure for a bank application. The following document pages are expected to be captured by the user during a mobile upload session for further processing in a transaction:

1. The first page of the passport.
2. All pages of the application.
3. Proof of residence, which includes pages of the rental agreement or pages of the utility bill.

▼ Show JSON sample

```
{
  "version": "2.4",
  "elements": [
    {
      "type": "standalone",
      "document": {
        "name": "Passport",
        "type": "passport",
        "count": "1",
        "galleryEnabled": false,
        "pages": [
          {
            "name": "First page with photo",
            "count": "1",
          }
        ]
      }
    },
    {
      "type": "standalone",
      "document": {
        "name": "Application",
        "type": "document",
        "count": "1",
        "galleryEnabled": true,
        "pages": [
          {
            "name": "Page",
            "count": "any",
          }
        ]
      }
    },
    {
      "type": "single_choice",
      "name": "Proof of residence",
      "documents": [
        {
          "name": "Rental agreement",
          "type": "document",
          "count": "1",
          "galleryEnabled": true,
          "pages": [
            {
              "name": "Page",
              "count": "any",
            }
          ]
        },
        {
          "name": "Utility bill",
          "type": "document",
          "count": "2",
          "galleryEnabled": true,
          "pages": [
            {
              "name": "Page",
              "count": "any",
            }
          ]
        },
      ]
    }
  ]
}
```

Tracking document processing and downloading results

You can track the status of your transaction and download the file containing the processing results.

Send a **GET** request to the **transactions/<transaction-id>** resource:

```
GET https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-id
```

Run the following command:

▼ For Windows

```
curl -X GET https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-id \
-H "Authorization: Bearer token"
```

▼ For Linux

```
curl -X GET https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-id \
-H 'Authorization: Bearer token'
```

Sample command results

```
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "status": "Processing",
  "manualReviewLink":"https://vantage.abbyy.com/api/publicapi/v1/verify?
documentIds=9838448d-72ae-4e9a-b071-2bb16f732e46",
  "sourceFiles": [
    {
      "id": "7b2eed6f-3fdd-43b9-a178-7211d0a8d5bb",
      "name": "Invoice07.JPG"
    }
  ]
}
```

The **status** key can have one of the following values:

- **Processing** means that the document is currently being processed.
- **Processed** means that processing has completed and the results can be downloaded.

To download a file containing the processing results, send a **GET** request to the **transactions/<transaction-id>/files/<file-id>/download** resource and specify the IDs of your transaction and document file:

```
GET https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-id/files/file-
id/download
```

Note: The file ID can be found in the **sourceFiles** property in the results for the transaction status request.

Run the following command:

For Windows

```
curl -X GET "https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-
id/files/file-id/download" \
-H "Authorization: Bearer token"
```

For Linux

```
curl -X GET 'https://vantage.abbyy.com/api/publicapi/v1/transactions/transaction-
id/files/file-id/download' \
-H 'Authorization: Bearer token'
```

The command will return the document processing results in JSON format.

Note: You can also download the results in PDF format. To do so, select the Output activity, click **Settings** in the **Exported Data** section in the **Actions** pane, and choose PDF as your preferred format.

Integrating Vantage Mobile Input to native applications

Vantage Mobile Input can be integrated into native iOS applications.

The purpose of Vantage Mobile Input is to simplify capturing and uploading documents directly from a mobile device to ABBYY Vantage, guiding users through the capture process according to predefined types and numbers of documents. Vantage Mobile Input will automatically access the quality of captured images and filter out low-quality photos not suitable for OCR. Images will then be cropped and justified for further processing by ABBYY Vantage skills.

Vantage Mobile Input can operate in two modes:

- **Connected mode** designed for real-time applications that require instant data processing. Captured images are immediately uploaded into ABBYY Vantage. This mode requires an active Internet connection.
- **Disconnected mode** designed for uninterrupted document capture even when no Internet connection is available. Captured document images are stored on the device and uploaded to ABBYY Vantage once the device is connected to the Internet.

This section provides instructions for integrating Vantage Mobile Input, including the user interface, into your iOS application.

- Step-by-step guides to the main scenarios
- Swift API reference

Step-by-step guides to the main scenarios

In this section you can find detailed "How to..." instructions for each mode.

- How to add the library to your Xcode project
- How to capture and upload documents in connected mode
- How to capture and save documents in disconnected mode

How to add the library to your Xcode project

To integrate Vantage Mobile Input into your application, you will need to add the library to your project, copy the resource files, and sign the framework.

1. Copy the **VantageMobileInputSDK** distribution package to your project folder.
2. On the **Build Options** tab, set the **Enable Bitcode** option to **No**.
3. Select your project in the **Target** group and open the **General** tab. Drag VantageMobileInput.framework from VantageMobileInputSDK/libs/ in Finder to the **Framework**, **Libraries** and **Embedded content** section. Set the **Status** of VantageMobileInput.framework to **Do not Embed**.
4. Add a **Run Script** phase to copy all the frameworks to the application bundle:

    a. In **Build Phases**, add a new **Run Script** phase.

b. Specify the path to the **copy_frameworks.sh** script located in the **libs/** folder:

```
/bin/sh "${SRCROOT}/VantageMobileInputSDK/libs/copy_frameworks.sh"
```

This will also remove from the frameworks any non-active CPU architectures (the exact list depends on the project settings) and sign the resulting framework.

5. Add a **Run Script** phase to copy the resource files and set up the copying rules:

   a. In **Build Phases**, add a new **Run Script** phase.

   b. Specify the path to the **copy_assets.py** script located in the **assets/** folder:

```
python3 "${SRCROOT}/VantageMobileInputSDK/assets/copy_assets.py"
```

6. Add the NSCameraUsageDescription and NSPhotoLibraryUsageDescription keys into the **info.plist** file to request access to the device's camera and to the user's photo gallery.

How to capture and upload documents in connected mode

For applications that work with an active Internet connection, use connected mode.

To launch Vantage Mobile Input in your application, perform the following steps:

1. Create an instance of MobileInputParameters using the createMobileInputParameters method with the mobile input link.

Note: For instructions on how to get a mobile input link, see the Vantage Mobile Input with the Vantage API section.

2. Create a scanMode enumeration with the **scanAndUpload** option and the MobileInputParameters instance to launch the connected mode.

3. Initiate a mobile input session and display the user interface:

   a. To start a new mobile input session in connected mode, use the scanDocuments method of the MobileInputDocuments singleton instance with scanMode.

   b. To continue a mobile input session, use the continueScanningDocuments method of the MobileInputDocuments singleton instance with scanMode and **SessionIdentifier**. To get a list of all started sessions use the savedSessions variable.

4. Check the operation result in the CaptureCompletion handler, which will contain one of the following:

   a. **SessionIdentifier** if the user successfully uploaded documents to the transaction.

   b. NSError if the user canceled the capture session, or if any error occurred. A canceled mobile input session can be continued and uploaded later.

If the upload is successful, the mobile input session will be removed.

How to capture and save documents in disconnected mode

For applications that work without an active Internet connection, use disconnected mode. This mode allows the capture and storage of documents with predefined structure. When an Internet connection becomes available, documents can be uploaded to ABBYY Vantage by switching to connected mode.

To launch Vantage Mobile Input in your application, perform the following steps:

1. Create an instance of MobileInputParameters.documentsToScan to define the types and number of documents to be uploaded during a mobile input session.
2. Create a scanMode enumeration with the **scan** option and the MobileInputParameters.documentsToScan instance to launch the disconnected mode.
3. Initiate a mobile input session and display the user interface:
   a. To start a new mobile input session in disconnected mode use the scanDocuments method of the MobileInputDocuments singleton instance with scanMode.
   b. To continue a mobile input session, use the continueScanningDocuments method of the MobileInputDocuments singleton instance with scanMode and **SessionIdentifier**. To get a list of all started sessions, use the savedSessions variable.
4. Check the operation result in the CaptureCompletion handler, which will contain one of the following:
   a. **SessionIdentifier** if the user successfully completed the mobile input session.
   b. NSError if any error occurred.

For uploading documents, you need to switch to connected mode and use the uploadDocuments method:

1. Create an instance of MobileInputParameters using the createMobileInputParameters method with the parameters from the mobile input link.

Note: The expiration period for the mobile input link is limited. Before creating an instance of MobileInputParameters obtain a new mobile input link. See the Vantage Mobile Input with the Vantage API section for details.

2. Use the uploadDocuments method with the **SessionIdentifier**, MobileInputParameters, and CaptureCompletion.

Note: Unused mobile input sessions can be removed using the deleteSession method. To get a list of all started sessions use the savedSessions variable.

Code samples

The Vantage Mobile Input distribution package includes the code sample that shows how to use API and provide examples of typical scenarios.

The code sample can be found in the root folder of the distribution package. The sample is provided in Swift.

Folder name	Description
VantageMobileInputSDKSampleSwift	This is a UIKit-based demo application demonstrating how to use Vantage Mobile Input in connected and disconnected mode using the API.

Configuring the code sample

The sample can be opened and built directly from where it is located in the downloaded distribution package. To configure the code sample, change the bundle ID before building, modifying or otherwise using the sample.

API reference

This section describes the Swift API within the Vantage Mobile Input SDK library, which allows you to integrate the Vantage Mobile Input feature into your iOS application.

- MobileInputDocuments class
- MobileInputParameters structure

MobileInputDocuments class

The **MobileInputDocuments** class is the main Vantage Mobile Input SDK class serving to launch Vantage Mobile Input in the application.

Note: Use shared instances of the singleton **MobileInputDocuments** class.

```
class MobileInputDocuments {
    typealias TransactionID = String
    typealias CaptureProgress = (_ progress: Float) -> Void
    typealias SessionIdentifier = String
    typealias CaptureCompletion = (Result<SessionIdentifier, NSError>) -> Void

    static var shared: MobileInputDocuments
```

TypeAlias

Type	Description
TransactionID	The ABBYY Vantage transaction identifier.
SessionIdentifier	The mobile input session identifier.
CaptureProgress	The upload progress handler (value between 0 and 1).
CaptureCompletion	The mobile input session completion handler with the operation result.

Methods

Method	Description
createMobileInputParameters	Creates an instance of MobileInputParameters with the mobile input link in connected mode.
scanDocuments	Starts a new mobile input session.
continueScanningDocuments	Continues the mobile input session.
uploadDocuments	Uploads captured documents to the transaction.
deleteSession	Deletes all stored information for the mobile input session.

captureCompletion

The **CaptureCompletion** closure handles the result obtained from scanDocuments, continueScanningDocuments, or uploadDocuments methods.

```
        switch result {
             case let .success(sessionIdentifier):
                    //Handle successful upload
             case let .failure(error):
                    //Handle the error
          }
```

Possible results

Result	Parameter	Condition	Action
.success(sessionIdentifier)	sessionIdentifier	The user has either (a) successfully uploaded documents to the transaction in connected mode or (b) successfully completed a mobile input session in disconnected mode.	Returns the **SessionIdentifier**.
.failure(error)	NSError	The user has canceled the capture session, or an error has occurred during the mobile input session.	Returns the **error**.

createMobileInputParameters method

The **createMobileInputParameters** method creates and returns an instance of MobileInputParameters for connected mode based on a provided mobile input link.

Note: All mobile input links eventually expire. Before creating an instance of the MobileInputParameters, obtain a new mobile input link. For information on how to obtain a mobile input link, see the Vantage Mobile Input with the Vantage API section.

```
func createMobileInputParameters(fromMobileInputLink mobileInputLink: URL) throws ->
MobileInputParameters
```

Parameters

Parameter	Description
mobileInputLink	The mobile input link.

Return values

An instance of the MobileInputParameters for capturing and uploading documents in connected mode.

▼ Example

```
do {
     let mobileInputParameters = try
MobileInputDocuments.shared.createMobileInputParameters(fromMobileInputLink: url)
     return mobileInputParameters
   } catch {
      //Handle the error
     }
```

scanDocuments method

The **scanDocuments** method initiates a new mobile input session in connected or disconnected mode. This method presents user interface components.

```
func scanDocuments(scanMode: ScanMode,
                          presenter: UIViewController,
                          completion: @escaping CaptureCompletion) ->
SessionIdentifier
```

Parameters

Parameter	Description
scanMode	The mode in which the mobile input session is launched: connected or disconnected.
UIViewController	The view controller which displays user interface components.

Parameter	Description
CaptureCompletion	The mobile input session completion handler which receives the operation result.

Return values

The mobile input session identifier.

Example

```
let sessionIdentifier = MobileInputDocuments.shared.scanDocuments(scanMode:
scanMode, presenter: self) { result in
      switch result {
           case .success(let sessionIdentifier):
           //The operation has been successfully completed.
           case .failure(let error):
           //The operation has failed.
           }
      }
```

continueScanningDocuments method

The **continueScanningDocuments** method continues the mobile input session with the scpecified **SessionIdentifier**. This method presents user interface components.

```
func continueScanningDocuments(scanMode: ScanMode,
                                  sessionIdentifier: SessionIdentifier,
                                  presenter: UIViewController,
                                  completion: @escaping CaptureCompletion)
```

Parameters

Parameter	Description
scanMode	The mode in which the mobile input session is launched: connected or disconnected.
SessionIdentifier	The mobile input session identifier.
UIViewController	The view controller which displays user interface components.
CaptureCompletion	The mobile input session completion handler with the operation result.

Example

```
mobileInputDocuments.shared.continueScanningDocuments(scanMode: scanMode, presenter:
self, sessionIdentifier: sessionInfo.sessionIdentifier, completion: { result in
      switch result {
           case .success(let sessionIdentifier):
           //The operation has been successfully completed.
           case .failure(let error):
           //The operation has failed.
           }
      }
```

uploadDocuments method

The **uploadDocuments** method uploads captured documents to ABBYY Vantage.

```
func uploadDocuments(mobileInputParameters: MobileInputParameters,
                     sessionIdentifier: SessionIdentifier,
                     progressHandler: CaptureProgress?,
                     completion: @escaping CaptureCompletion)
```

Parameters

Parameter	Description
MobileInputParameters	An instance of **MobileInputParameters**.
SessionIdentifier	The mobile input session identifier.
CaptureProgress	The upload progress handler (value between 0 and 1).
CaptureCompletion	The mobile input session completion handler with the operation result.

Example

```
MobileInputDocuments.shared.uploadDocuments(mobileInputParameters:
mobileInputParameters, sessionIdentifier: sessionInfo.sessionIdentifier,
progressHandler: { progress in
       //Handle the uploading progress
       },
  completion: { result in
       switch result {
            case .success(let sessionIdentifier):
            //The operation has been successfully completed.
            case .failure(let error):
            //The operation has failed.
       }
```

deleteSession method

The **deleteSession** method removes all stored information for the mobile input session with the specified **sessionIdentifier**.

```
@discardableResult
func deleteSession(_ sessionIdentifier: SessionIdentifier) -> NSError?
```

Parameters

Parameter	Description
SessionIdentifier	The mobile input session identifier.

Return values

Returns **nil** if the mobile input session is successfully removed. Returns **NSError** if the mobile input session is in use.

▾ Example

```
MobileInputDocuments.shared.deleteSession(id: sessionInfo.sessionIdentifier)
```

MobileInputParameters structure

The **MobileInputParameters** structure is used to capture and upload documents in connected mode. It is created by using the createMobileInputParameters method with the mobile input link.

Note: All mobile input links eventually expire. Before creating an instance of **MobileInputParameters**, obtain a new mobile input link. For information on how to obtain a mobile input link, see the Vantage Mobile Input with the Vantage API section.

```
struct MobileInputParameters {
        let transactionId: String
        let documentsToScan: MobileInputParameters.DocumentsToScan
    }
```

Parameters

Parameter	Description
transactionId	The ABBYY Vantage transaction identifier.
documentsToScan	The document schema with the types and number of documents and document pages to be captured by the user during the mobile input session.

DocumentsToScan enum

The **DocumentsToScan** enumeration defines a document schema with specific types and number of documents and document pages to be captured during the mobile input session.

```
struct MobileInputParameters {

            enum DocumentsToScan {
            case webUrl(URL)
            case jsonData(Data)

            static let idCard: DocumentsToScan
            static let passport: DocumentsToScan
            static let multipleDocuments: DocumentsToScan
            static let singleDocument: DocumentsToScan
}
```

Values for **DocumentsToScan** can be initialized in two ways:

- Using the JSON document schema
- Using one of the static properties

JSON document schema

To configure the documents schema to fit a specific user scenario, use a JSON format. Specify the number, types, and names of documents, and the number and names of pages of each document the user should capture. See this section on how to create a document schema in the JSON file.

To initialize the value of the **DocumentsToScan** type with a JSON file, do one of the following:

- Specify the encoded URL with the direct access link to the JSON file as a value of the **webUrl** parameter.

▾ Example

```
documentsToScan =
MobileInputDocuments.MobileInputParameters.DocumentsToScan.webUrl(url)
```

- · Specify the JSON data as a value of the **jsonData** parameter.

▾ Example

```
documentsToScan =
MobileInputDocuments.MobileInputParameters.DocumentsToScan.jsonData(data)
```

Static properties

Standard static properties allow the creation of a document structure based on a single document type. The specified document type will be captured during the mobile input session.

To initialize the value of the **DocumentsToScan** type for the capture of a specific document type, specify one of the static properties.

▾ Example

```
documentsToScan =
MobileInputDocuments.MobileInputParameters.DocumentsToScan.passport
```

Property	Description
idCard	Captures both sides of an ID card.
passport	Captures the first page of a passport.
multipleDocuments	Captures multiple documents of any type.
singleDocument	Captures a single document of any type.

sessionInfo structure

The **SessionInfo** structure contains parameters for the mobile input session.

```
struct SessionInfo {
    let sessionIdentifier: SessionIdentifier
    let isReadyForUpload: Bool
    let capturedDocumentsCount: Int
    let capturedPagesCount: Int
    let createdAt: Date
    let thumbnail: URL?
    let thumbnailOrentation: UIImage.Orientation?
}
```

Parameters

Parameter	Description
SessionIdentifier	The mobile input session identifier.
isReadyForUpload	This flag indicates whether document can be sent to the transaction. Depends on whether all the required documents have been captured in the current mobile input session.
capturedDocumentsCount	The number of documents in which all required pages have been captured.
capturedPagesCount	The number of captured pages.
createdAt	The date when the mobile input session has been created.
thumbnail	The thumbnail of the first page of the first document.
thumbnailOrientation	The thumbnail orientation.

scanMode enum

The **scanMode** enumeration determines the mode in which the mobile input session will be launched.

```
enum ScanMode {
    case scanAndUpload(mobileInputParameters: MobileInputParameters)
    case scan(documents: MobileInputParameters.DocumentsToScan)
}
```

Note: All mobile input links eventually expire. Before creating the **scanMode** with the **scanAndUpload** option for the mobile input session in connected mode, you need to obtain a new mobile input link. For information on how to obtain a mobile input link, see the Vantage Mobile Input with the Vantage API section.

Options

Option	Description
scanAndUpload	Launches the mobile input session in connected mode.
scan	Launches the mobile input session in disconnected mode.

Associated Values

Value	Description
MobileInputParameters	An instance of **MobileInputParameters**.
MobileInputParameters.DocumentsToScan	An instance of **MobileInputParameters.DocumetsToScan**.

▾ Example

```
let scanMode: MobileInputDocuments.ScanMode = .scanAndUpload(mobileInputParameters:
mobileInputParameters)
```

savedSessions var

The **savedSessions** variable contains the data of all saved mobile input sessions.

```
var savedSessions: [SessionInfo] {}
```

Values

Value	Description
SessionInfo	Contains the parameters of the mobile input session.

Example

```
let sessions = MobileInputDocuments.shared.savedSessions
```

Distribution kit

The Vantage Mobile Input distribution package includes the libraries, the resource files, and the sample.

Folder	File name	Description
VantageMobileInput/assets/patterns	CnnCrop.Classifier.cnnmodel CnnCrop.Detector.Generic.cnnmodel CnnCrop.Detector.GenericWithID.cnnmodel CropClassifierGeneric.imodel CropClassifierPhoto.imodel DIQBlockClassifier.imodel DIQClassifier.imodel FastCrop.imodel	The resource files which required for image capture scenario.
VantageMobileInput/libs	AbbyyUI.framework	The framework for integrating the native interface for image capture to the application.
	Mobile.Imaging.framework FineObj.framework Mobile.SDK.framework VantageMobileInput.framework NeoMathEngine.framework NeoML.framework PortLayer.framework	The Vantage Mobile Input SDK frameworks.
VantageMobileInputSDKSampleSwift	All files in the folder.	This sample illustrates the steps you need to perform to add Vantage Mobile Input to your application.
	VantageMobileInput_DevelopersGuide.pdf	The Developer's Guide.

Device requirements

iOS version: 15.x or later.

Recommended devices:

- iPhone 11, iPad Pro or later versions.

Third-party trademarks

Third-Party Software/Open-Source Software Used in

ABBYY Vantage Mobile Input SDK, version 1.0.

This document contains licenses and notices for free/open-source software and third-party software used in the Software identified above. If you have any questions with respect to the free/open-source software and third-party software listed in this document, please contact us at legaloperations@abbyy.com.

Please include the following reference number in your requests: ABBYY Vantage Mobile Input SDK 1.0. – 22042024

Licensed under the Apache-2.0 License:

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0, and for your convenience, the License text is cited below.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a crossclaim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the

Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Component	Copyright
Kotlin	Copyright 2000-2018 JetBrains s.r.o. and Kotlin Programming Language contributors Copyright 2010-2023 JetBrains s.r.o and respective authors and developers

Component	Copyright
Kotlin Coroutines Core	Copyright 2000-2020 JetBrains s.r.o. and Kotlin Programming Language contributors Copyright 2016-2024 JetBrains s.r.o and contributors
Kotlin Coroutines Android	Copyright 2000-2020 JetBrains s.r.o. and Kotlin Programming Language contributors Copyright 2016-2024 JetBrains s.r.o and contributors
Android SQLite support library	Copyright (C) 2017-2024 requery.io Copyright (C) 2005-2012 The Android Open Source Project
Material Components for Android	Copyright (C) 2021 The Android Open Source Project
Insetter	Copyright 2019 Google LLC.
OkHttp	Copyright (C) 2019 Square, Inc.
Timber	Copyright 2013 Jake Wharton
Hilt	Copyright 2012 The Dagger Authors
Exifinterface	Copyright 2018 The Android Open Source Project

Licensed under other licenses:

Component	Copyright
Glide	License for everything not in third_party and not otherwise marked: Copyright 2014 Google, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY GOOGLE, INC. ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

Component	Copyright
	THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL GOOGLE, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Google, Inc. -- License for third_party/disklrucache: Copyright 2012 Jake Wharton Copyright 2011 The Android Open Source Project Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. -- License for third_party/gif_decoder: Copyright (c) 2013 Xcellent Creations, Inc.

Component	Copyright
	Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. -- License for third_party/gif_encoder/AnimatedGifEncoder.java and third_party/gif_encoder/LZWEncoder.java: No copyright asserted on the source code of this class. May be used for any purpose, however, refer to the Unisys LZW patent for restrictions on use of the associated LZWEncoder class. Please forward any corrections to kweiner@fmsware.com. -- License for third_party/gif_encoder/NeuQuant.java Copyright (c) 1994 Anthony Dekker NEUQUANT Neural-Net quantization algorithm by Anthony Dekker, 1994. See "Kohonen neural networks for optimal colour quantization" in "Network: Computation in Neural Systems" Vol. 5 (1994) pp 351-367. for a discussion of the algorithm.

Component	Copyright
	Any party obtaining a copy of these files from the author, directly or indirectly, is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

Information in this document is subject to change without notice and does not bear any commitment on the part of ABBYY.