ABBYY FlexiCapture 12 SDK, Release 1 Update 2: Release Notes

Table of Contents

1 Introduction

1.1 About This Document

This document contains technical information about Release 1 Update 2 of the ABBYY FlexiCapture 12 SDK.

1.2 About the Product

The ABBYY FlexiCapture 12 SDK is a powerful data capture software development kit for ISVs, OEM vendors, and service providers.

The ABBYY FlexiCapture 12 SDK is intended for developing data capture solutions for automated processing of invoices, medical forms, insurance claims, questionnaires, examination papers, ID cards, and other types of documents. It can also be used for creating customized data capture applications for various scanning devices and terminals.

The ABBYY FlexiCapture 12 SDK combines technologies and tools for processing documents with complex layouts and for document classification, data verification, and export to backend processing and archiving systems, all in a single development environment.

2 About the Release

2.1 Purpose of the Release

This is an update for the first release of the ABBYY FlexiCapture 12 SDK. The purpose of this release is to make the following features available to developers:

- Ability to load document definitions into the project
- Machine learning for processing invoices with different layouts from the same vendor
- PO numbers, BU and Vendor names extraction from invoices based on neural networks
- PO number extraction training on user's side
- Improved detection of line items with multi-line description
- Working with Online licenses using proxy servers
- Online product documentation
- XLSX export format support
- Support of Windows Server 2019
- Demonstration of FC SDK usage in Python

For a detailed description of these features, see the "Main Features" section.

2.2 Component Versions

Part #	1383/10
Build #	12.0.14.13
OCR Technology build #	16.1.814.24
APDFL version	15.0.4PlusP5e

2.3 Specifications

For the specifications, please refer to the document named "ABBYY *FlexiCapture 12 SDK, Release 1 Update 2: Specifications*".

3 Installing the ABBYY FlexiCapture 12 SDK

For the installation instructions, please refer to the FlexiCapture 12 SDK System Administrator's Guide, which you can find in the distribution package.

4 Compatibility

4.1 Compatibility with ABBYY FlexiCapture

The ABBYY FlexiCapture 12 SDK is compatible with projects created in ABBYY FlexiCapture 12 Update 1 (build 12.0.3.2634) or earlier). Projects created in the ABBYY FlexiCapture 12 SDK can be used in ABBYY FlexiCapture 12 Update 1 (build 12.0.3.2634 or later). You can also transfer projects to and from the Developer's Package shipped with the ABBYY FlexiCapture 12 SDK.

4.2 Licensing

Before you can start working with the ABBYY FlexiCapture 12 SDK, Release 1, Update 2, you will need to enter a serial number and activate your copy of the SDK.

For more information, see the "Licensing" section in the User Manual.

4.3 Upgrading from the ABBYY FlexiCapture Engine

Before upgrading, be sure to save backup copies of your existing projects.

Binary Incompatibility

You will need to recompile your application regardless of which version of the Engine you have previously used.

Initializing the ABBYY FlexiCapture 12 SDK

To initialize the ABBYY FlexiCapture 12 SDK, use the *GetEngineLoader* function (i.e. the same function that you used to initialized the ABBYY FlexiCapture Engine).

For more information about initializing the SDK, see this section of the User Manual: *Guided Tour > Tutorial > Getting Started with ABBYY FlexiCapture Projects > Step 1: Load the ABBYY FlexiCapture 12 SDK.*

Using ABBYY FlexiCapture projects

The ABBYY FlexiCapture 12 SDK can work with projects created in any of the following products:

- ABBYY FlexiCapture 12 SDK, Release 1, Update 2 (build 12.0.14.13)
- ABBYY FlexiCapture 12 Developer's Package shipped with ABBYY FlexiCapture 12 SDK, Release 1, Update 2, or earlier
- ABBYY FlexiCapture 12 (build 12.0.3.2634 or earlier)

Projects created in ABBYY FlexiCapture products *newer* than the ones listed above cannot be used with the ABBYY FlexiCapture 12 SDK.

To be able to use projects created in ABBYY FlexiCapture products *older* than the ones listed above, you will need to call the *Engine::UpdateProject()* method.

For more information, see this section of the User Manual: *Guided Tour > Tutorial*

No Support for the ABBYY FlexiCapture Processor

The ABBYY FlexiCapture 12 SDK is not compatible with the ABBYY FlexiCapture Processor. However, the SDK can be used with projects created in compatible versions of ABBYY FlexiCapture products (see above). To migrate a project from the Engine to the SDK, do the following:

1. Create a new ABBYY FlexiCapture project — either in the SDK by calling the *CreateProject* method of the *Engine* object, or in the Developer's Package shipped together with the SDK.
2. Using the Developer's Package, import your Document Definitions to the ABBYY FlexiCapture project created in step 1.
3. Create a working batch inside your ABBYY FlexiCapture project.
4. Open your ABBYY FlexiCapture project in the ABBYY FlexiCapture 12 SDK by using the *OpenProject* method of the *Engine* object.

5. To add images to your project, use *Batch::AddImage()*instead of calling the *FlexiCaptureProcessor::AddImage()* method.
6. To recognize images, use *Batch::Recognize()* instead of calling the *FlexiCaptureProcessor::RecognizeNextDocument()* method.

5 Main Features

5.1 Release 1

5.1.1 Data Capture for All Document Types

The ABBYY FlexiCapture 12 SDK is a delivery option for ABBYY FlexiCapture 12 that provides access to all the capabilities of ABBYY FlexiCapture by means of an API.

A data capture workflow in ABBYY FlexiCapture looks as follows:

The API allows you to pass a document through all of the stages. Additionally, it allows you to fine-tune each of the steps in accordance with your customer's scenario.

5.1.1.1 Obtaining Document Images

5.1.1.1.1 Adding Images

Data capture starts with adding new document images — either by scanning hard-copy documents or by taking existing digital images stored in files or memory. The ABBYY FlexiCapture 12 SDK provides an image import API for importing images from supported sources.

- To add images from a scanning device, use the *ScanManager* object, which exposes a set of properties and methods required to perform scanning.
- To add images from files, use the *AddImage()* and *AddImages()* methods of the *Batch* object.
- To add images directly from memory, use the *AddImageFromStream()* method of the *Batch* object.

For more information about adding images, see this section of the User Manual: *Guided Tour > Tutorial > Getting Started with ABBYY FlexiCapture Projects > Step 3: Add Images*.

5.1.1.1.2 Preprocessing Images

When the ABBYY FlexiCapture 12 SDK receives images, it performs a range of image preprocessing operations to improve image quality for further recognition or archiving. Images can be preprocessed either when they are added into a batch or prior to data extraction.

To specify preprocessing operations that should be applied to images upon loading, use the *ImageLoadingParams* object. The following image preprocessing operations are available:

- automatically correct image resolution
- overwrite images resolution
- correct image skew based on black separators, black squares, or text
- convert images to black and white
- rotate, invert, or mirror images
- reduce ISO noise
- remove color marks and other unwanted marks
- automatically crop images

You can also apply a preconfigured image enhancement profile by using the *SetImageEnhancementProfile* method of the *ImageLoadingParams* object (images will be preprocessed upon loading) or by using the *ApplyImageEnhancementProfile* method of the *Document* object (images will be processed after they have been loaded).

For more details about image enhancement profiles in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/image_enhancement_profiles.

For detailed instructions on managing image enhancement profiles, see this section of the User Manual: *API Reference > Objects > Image-Related Objects > ImageEnhancementProfile Object*.

5.1.1.2 Document Classification

The ABBYY FlexiCapture 12 SDK allows you to sort incoming documents into predefined categories using a preconfigured classifier. The classification technology is highly flexible and can be used to distinguish between:

- document types (e.g. you can separate passports from invoices)
- document variations (e.g. you can detect invoices received from different vendors or driver's licenses issued in different states)
- document boundaries (e.g. you can detect the first and subsequent pages of documents of different types)

A trained classifier processes documents by determining the class of each page. This information is then used during recognition to select and apply matching Document Definitions and to select the appropriate section variants; it is also possible to get page classes and use them directly.

For more details about the classification technology used in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/classify_intro.

The ABBYY FlexiCapture 12 SDK provides API methods and interfaces that allow you to:

- Use preconfigured classifiers in ABBYY FlexiCapture projects by using the *ClassificationBatch* and *UseClassifier* properties of the *BatchTypeParams* object.
- Train a new classifier from scratch (see the Advanced Classification section for details).[1]
- Train a classifier using feedback from verification operators to improve classification accuracy (see the "Training Based on Feedback" section for details).

For detailed instructions, see this section of the User Manual: *API Reference > Objects > Classification Training Objects*.

5.1.1.3 Data Extraction

Once the images are loaded into the ABBYY FlexiCapture 12 SDK, data extraction begins. In many data capture scenarios, data have to be extracted only from certain fields. The ABBYY FlexiCapture 12 SDK can be used to detect such fields on document images.

For this purpose, the SDK uses special formalized descriptions called *Document Definitions*. Document Definitions can be created either using the Developer's Package or directly via the API. The SDK can find data fields using the following information: relations between fields and other objects on the page, field contents, field size, lines drawn around fields, etc.

For more details about Document Definitions, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/templates_main.

To extract data by means of the ABBYY FlexiCapture 12 SDK, call the *Recognize()* method of the *Batch* or *Project* object.

[1] You can also train a classifier using the GUI of the Developer's Package.

5.1.1.4 Data Verification

Whenever ABBYY FlexiCapture is not sure whether a character has been recognized correctly or not, it marks it as unreliably recognized. Additionally, various types of other errors can be detected by means of validation rules. All such errors must be corrected by a verification operator, who will check the recognition results against the original image.

The ABBYY FlexiCapture 12 SDK provides tools for data verification.

For detailed instructions, see this section of the User Manual: *Guided Tour > Advanced Techniques > Verifying Recognized Data*.

Verification results can also be used for training a classifier (see the "Training Based on Feedback" section for details).

5.1.1.5 Data Export

Captured data can be exported to backend applications or converted into searchable PDF or PDF/A files for archiving purposes. Special technologies are used to export data to structured formats (like XML or CSV) and to various storage locations. For some formats, the images of the original documents can also be saved.

You can use the *Export()* method of the *Batch* or *Project* object to save document processing results to a file, and the *ExportToMemory()* method of the *Batch* or *Project* object to save document processing results to memory (see the "Export to Memory" section for details). Additionally, the following export capabilities are available:

- Redaction, which obliterates confidential or sensitive data

- MRC compression, which reduces the size of PDF files

For more information about data verification, see this section of the User Manual: *Guided Tour > Tutorial > Getting Started with ABBYY FlexiCapture Projects > Step 5: Export Results*.

5.1.1.5.1 Export to Memory

Export to memory can be used to speed up document processing, as documents will not be written to disk and their data will be accessible directly from memory. For example, data extracted from a mortgage application can be passed to another program in order to calculate risks when deciding on a loan. Also, some companies with strict data protection policies (e.g. financial and healthcare organizations) prohibit applications from writing certain information to their hard disks. In such cases, all document processing must be done in virtual memory.

During export, one or multiple files can be created. Files can be placed into folders based on export settings, forming a hierarchical structure in a virtual file system. You can access these files through the *VirtualFileSystemNode* object, which is a return value of the *ExportToMemory()* method of the *Batch* or *Project* object.

A virtual node (*VirtualFileSystemNode*) may have child items. To get a list of child items for a virtual node, use the *ChildItems* property. To get the contents of a file stored in a virtual file system, use the following methods and properties of the *VirtualFileSystemNode* object:

- *IsRoot, IsDirectory* – Specify whether the node is a root of the virtual file system or a file.
- *Name* – Stores the name of a file or directory.

- *Size* – Stores the size of a node in bytes. Available for files only.
- *SaveToStream()* – Saves the contents of a file to the stream.
- *ReleaseMemory()* – Frees up memory occupied by export results.

5.1.2 Invoice Processing

The ABBYY FlexiCapture 12 SDK provides developers with technology for invoice processing, replacing labor-intensive manual data input with transparent, manageable, efficient, and automated data capture based on smart document analysis and optical character recognition. The invoice processing technology offers predefined settings, validation rules, and advanced database look-up specifically tailored for processing invoices.

The technology can identify data fields on most invoices, enabling you to start processing right away. The technology can also be trained based on feedback from verification operators to improve data extraction for invoices with non-standard layouts.

For more details about the invoice processing technology used in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/invoice_reader/ir_titletopic.

The ABBYY FlexiCapture 12 SDK allows you to:

- Set up an ABBYY FlexiCapture project for invoice processing.
- Use vendor and business unit data sets for more accurate data capture.
- Process invoices using configured ABBYY FlexiCapture projects as described in the "Data Extraction" section.
- Train the technology using verification results (see the "Training Based on Feedback" section for details).

The distribution package also includes sample code for invoice processing (see the "Sample Code Library" section for details).

5.1.2.1 Setting Up ABBYY FlexiCapture Projects for Invoice Processing

ABBYY FlexiCapture allows you to create a special type of project for processing invoices. There are 4 possible types of invoice processing projects intended for the following for 4 regions:

- Australia and New Zealand
- Canada
- EU
- USA

An invoice processing project can be created either via the GUI in the Developer's Package or directly via the API (see the "API-Based Setup" section for details).

Before you can start processing invoices, some adjustments to the default settings may be required. You can use the *InvoiceSettings* object (accessible through the *Document Definition* object) to set up the following:

- The *Countries* object, which is used to specify the country from which invoices originate.
  - You can also modify country-related settings (e.g. specific tax rates, currency formatting, regular expressions, etc.) using a *Country* object.
- The *InvoiceLanguages* object, which is used to specify the set of languages that will be used for recognition. Languages are also dependent on the invoice country.
- The *InvoiceFeatures* object, which is used to enable additional invoice processing features.
- The *DocumentStatus* object, which is used to access document statuses and their descriptions.

For more details about the setup options available in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/ir_howtoadjustproject.

You can also use data sets to improve the accuracy of data capture (see the "Using Data Sets for More Accurate Data Capture" section for details).

5.1.2.2 Using Data Sets for More Accurate Data Capture

To capture data from invoices, ABBYY FlexiCapture relies on information about vendors and business units. Vendors are companies that issue invoices, while business units are companies or divisions that receive invoices.

To find and apply a matching Document Definition to an invoice, ABBYY FlexiCapture will first look for vendor and business unit fields.

- Based on the information contained in the vendor field, ABBYY FlexiCapture determines the language of the invoice and the formatting rules used for of numbers, dates, and amounts.
- Based on the information contained in the business unit field, ABBYY FlexiCapture decides which rules to use for checking the data captured from the invoice.

For more details about the data sets used in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/ir_dbconnection.

There are two ways to inform ABBYY FlexiCapture about vendors and business units.

The first way is to link a database to a data set. A data set is a local copy of a table from an external database that stores information about vendors or business units. The values stored in a data set can be typified and normalized. Connection preferences can be specified in the Developer's Package. You can use the *UpdateFromDB()* method of the *DataSetTableRecords* object to periodically synchronize your data set with its external database.

The second way is to populate a data set manually using the *DataSetTableRecords* object, which offers the following methods:

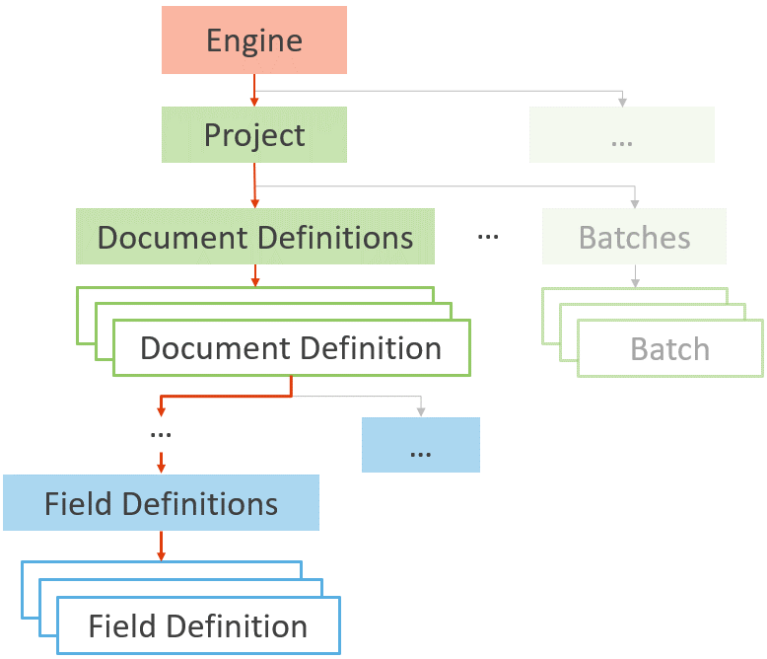
- *AddNew()* – Adds a new record to a data set.
- *DeleteAt()*, *DeleteAll()* – The first removes a specific record and the second removes all records from a data set.
- *Item()* – Provides access to a single record in a data set.
- *LookUp()* – Returns the results of a full-text search in a data set.

5.1.3 API-Based Setup

Some users may need to set up or modify ABBYY FlexiCapture projects without using the GUI of the Developer's Package. The following categories of users may want to use the API to set up their projects:

- Software developers who are working with document management systems and need to modify document metadata directly from their solutions
- Business process outsourcers who operate many FlexiCapture projects and need to modify the settings of specific fields automatically

The hierarchy of the API objects is shown in the figure below. *Engine* is the top object in the hierarchy. All other objects and methods are accessible via the *Engine* object.



The *Engine* object and objects that inherit from it are used to set up and fine-tune the data capture process. For example, you can set up invoice processing settings (see Setting Up FlexiCapture Projects for Invoice Processing) and populate data sets for more accurate invoice recognition (see Using Data Sets for More Accurate Data Capture).

Additionally, you can use the API to:

- Create an ABBYY FlexiCapture project (use the *CreateProject()* or *CreateInvoiceProject()* method of the *Engine* object).
- Create batch types and specify their processing parameters (use the *AddNew()* method of the *BatchTypes* object).
- Create Document Definitions (use the *AddNew*() method of the *DocumentDefinitions* object) and create fields in Document Definitions (use the *AddNew*() method of the *FieldDefinitions* object).
- Set up recognition parameters for fields (use the *RecognitionParams* property of the *FieldDefinition* object).

5.1.4 Advanced Classification

The classification technology included in the ABBYY FlexiCapture 12 SDK lets users classify documents into different classes (e.g. “driver’s license,” “bank statement,” “tax return,” “contract,” “invoice,” etc.) and document variations (e.g. invoices from different vendors) without developing sophisticated FlexiLayouts. It enables a mailroom scenario (i.e. automatic sorting of incoming documents) and simplifies the creation of FlexiLayouts by eliminating the need to define identification elements. It also allows creating separate FlexiLayouts for different document variations (e.g. you can have a separate FlexiLayout for each vendor).

Classification can use information from images and the textual contents of documents.

- Image-based classification uses convolutional neural networks and distinguishes between documents that differ visually.
- Text-based classification relies on statistical and semantic text analysis and requires OCR.

It is also possible to combine both approaches in two different classification modes – *Fast* and *Thorough*.

Image- and text-based classification is automatic, because to train a classifier, you only need to provide a sample set of documents classified into reference classes. No special rules are required, but you can still use rules to adjust the results of automatic classification.

You can also prioritize recall over precision (or the other way round) or use the “balanced” mode.

For more details about the classification technology used in ABBYY FlexiCapture 12, click this link: http://help.abbyy.com/en-us/flexicapture/12/distributed_administrator/classify_intro.

The API allows you to:

- Train a classifier from scratch.
- Use feedback from verification operators to improve the accuracy of a classifier (see the "Training Based on Feedback" section for details).

5.1.4.1 Training a Classifier from Scratch

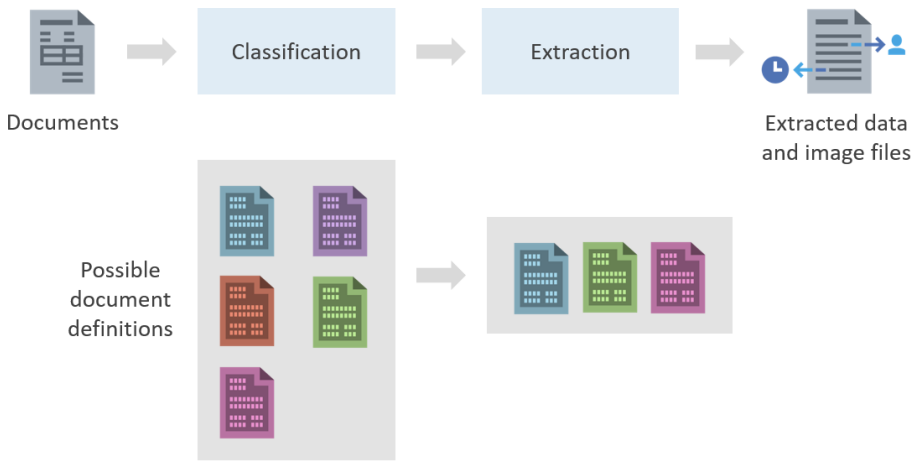
There are two ways to create a classifier:

- Using the GUI of the Developer's Package
- Using API methods

To train a classifier by means of the API, use classifier training batches, which are represented by *ClassificationTrainingBatch* objects. To train your classifier, follow these steps:

1. **Add document images to your training batch and mark up the data.**
   a) Using the *AddImages* method of the *Batch* object, add document images to your training batch.
   b) For each document in the training batch, specify whether it should be used *for training* or *for testing* the classifier. To do this, use the *TrainingState* property of the *ClassificationTrainingDocument* object. The recommended ratio of documents used for training to documents used for testing is 70:30 for each class.
   c) Using the *ReferenceClass* property of each *ClassificationTrainingPage* object in the training batch, assign page classes to the added document images.
2. **Map reference classes to Document Definition sections.**

Classification simplifies the detection of document types and document variations, reducing the number of potentially matching Document Definitions.



This is achieved by mapping reference classes to Document Definition sections using the *AddNew()* method of the *LinksToSections* collection of the *PageClass* object.

3. **Set up training parameters and run the training.**
   a) Get the *ClassificationTrainer* object from the classifier training batch.
   b) Using the *ClassificationTrainingParams* property of the *ClassificationTrainer* object, set the training parameters. Specify a classifier training profile (i.e. image-based, text-based, or combined) and a precision-to-recall ratio (or select "balanced" mode).
   c) Use the *Train()* method of the *ClassificationTrainer* object to start the training.
4. **Publish your classifier.**

Upon calling the *Train()* method, the ABBYY FlexiCapture 12 SDK provides classification statistics in the *TrainingResults* object. These statistics can be obtained via the *ValidationResults* object. To publish the classifier and disable the training mode, use the *CheckIn()* method of the *ClassificationTrainer* object.

The distribution package contains sample code which demonstrates an implementation of this scenario (see the Sample Code Library section for details).

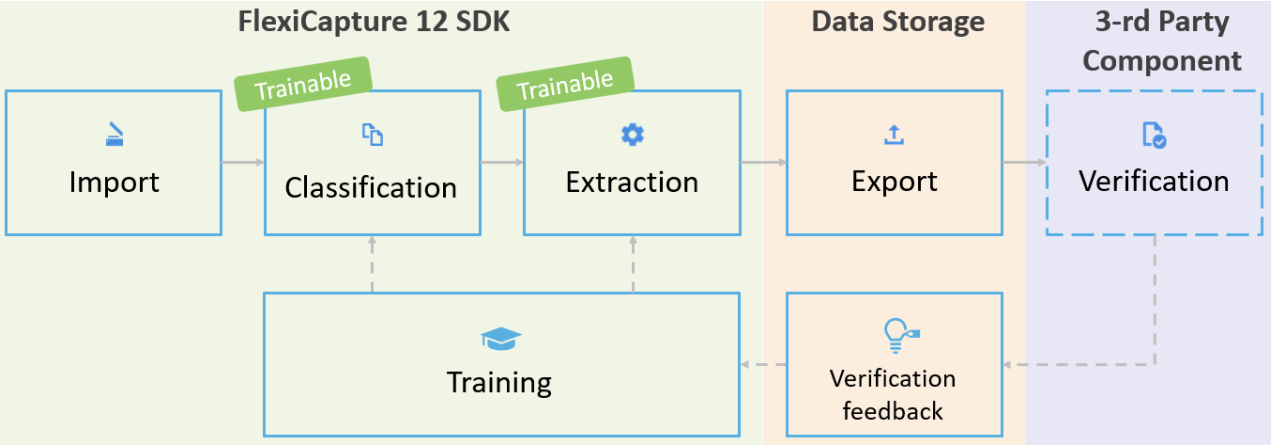
For more information about training classifiers, see this section of the User Manual: *Guided Tour > Tutorial > Document Classification Training*.

5.1.5 Training Based on Feedback

The ABBYY FlexiCapture 12 SDK allows you to implement training for field extraction and document classification. Training improves the accuracy of field extraction and document classification.

Field extraction and document classification can learn from feedback collected from verification operators. Feedback can be collected using a verification component, implemented by the developer, and should contain:

- Information about the correct field regions (for field extraction).
- Information about the correct image classes (for document classification).



5.1.5.1 Field Extraction Training

ABBYY FlexiCapture can be trained to detect the position of simple fields and of recurring groups of fields in structured or a semi-structured documents. This may be useful for:

- Preliminary project training, e.g. when fields have been created using the API and there are no rules for detecting field regions (see the "API-Based Setup" section for details).
- Automatic learning while processing documents.

Field extraction can be trained for ABBYY FlexiCapture projects designed to process invoices or for projects designed to process any other kinds of documents.

To train field extraction, use field extraction training batches, which are represented by *FieldsExtractionTrainingBatch* objects. Follow these steps:

1. Open or create a training batch.

Each training batch will be used to train fields either for a document section or its variant (in the case of a non-invoice project) or for a vendor (in the case of an invoice project). A variant corresponds to a record in the "Variants" data set associated with the document section and a vendor corresponds to a record in the "Vendors" data set associated with the invoice Document Definition.

To start the training process, open the field extraction training batch that you created for the given section, variant or vendor. Alternatively, create a *FieldsExtractionTrainingBatch* object using the *AddNew()* method of the *FieldsExtractionTrainingBatches()* object. As input parameters, pass the *SectionDefinition* object for the section and the *DataSetTableRecord* object for the section variant or vendor.

2. **Add verified document images to the training batch.**
   a) Using the *AddImages()* method of the *Batch* object, add verified document images to the training batch.
   b) Call the *SetFieldRegion()* method of the *Page* object to assign the correct region to each field to be trained.
3. **Train field extraction.**

To start the training, call the *Train()* method of the *FieldsExtractionTrainer* object. The Document Definition will be automatically updated with the training results.

For more information about field extraction training process, see this section of the User Manual: *Guided Tour > Tutorial > Field Extraction Training*.

The distribution package contains sample code which demonstrates an implementation of this scenario (see the Sample Code Library section for details).

5.1.5.2 Classifier Training

Classifiers can also learn from feedback collected from verification operators.

Training a classifier based on feedback is similar to training a classifier from scratch (see the "Training a Classifier from Scratch" section for details). An important difference is that you do not need to create a new classifier training batch — you only need to open the *ClassificationTrainingBatch* object that corresponds to the classifier to be trained. To train a classifier using feedback from verification operators, follow these steps:

1. **Add document images to the training batch and mark up the data.**
   a) Using the *AddImages* method of the *Batch* object, add document images with verified classes to the training batch.
   b) For each document in the training batch, specify that it should be used *for training* the classifier. To do this, use the *TrainingState* property of the *ClassificationTrainingDocument* object.
   c) Using the *ReferenceClass* property of each *ClassificationTrainingPage* object in the training batch, assign the correct classes to the added document images.
2. **Remap the classes to the Document Definitions, if required.**

When correcting the classification results, the verification operator may need to assign completely new classes to documents. If this is the case, you will need to remap the classes to the Document Definitions.

3. **Using the *ClassificationTrainingParams* property of the *ClassificationTrainer* object, set the training parameters and start the training.**
4. **Review the training results and publish your classifier.**

Upon calling the *Train()* method, the ABBYY FlexiCapture 12 SDK provides classification statistics in the *TrainingResults* object. These statistics are calculated both for the previously used classifier (*CurrentResults* object) and for the newly trained classifier (*ValidationResults* object). To calculate the statistics, the technology uses only those documents in the training batch that were marked *for testing.* The statistics are based on the classification results for all documents and show how well the classifier performs on the provided set of sample documents. The statistics include recall, precision, F-measure, and a confusion matrix to help you identify the most commonly confused classes.

By comparing the statistics for the old and the new classifiers, you can decide if the new classifier should be preferred. Alternatively, you can check out the *ShouldApply* property of the *TrainingResults* object, which is an internal estimate of whether the new classifier is better than the old one. Depending on your decision, call one of the following two methods:

- *CheckIn()* — This method will publish the classifier and switch off the training mode.
- *UndoCheckout()* — This method will discard the changes to the classifier and switch off the training mode.

The distribution package contains sample code which demonstrates an implementation of this scenario (see the "Sample Code Library" section for details).

For more about training classifiers, see this section of the User Manual: *Guided Tour > Tutorial > Document Classification Training*.

5.1.6 Processing Born-Digital Documents

Born-digital documents are documents that have been natively created in digital formats (e.g. by using office document authoring applications). The ABBYY FlexiCapture 12 SDK can processes born-digital documents in all popular office formats. All types of documents can be processed within the same workflow, without the need to separate documents in image formats form documents in text formats.

- Supported text formats: DOC, DOCX, RTF, HTML, TXT, ODT
- Supported worksheet formats: XLS, XLSX, ODS
- Supported presentation formats: PPT, PPTX, ODP

The following software can be used by the SDK to process born-digital documents:

- Microsoft Office 2007 or later (its use must be allowed in the SDK settings and a valid login and password must be provided)
- LibreOffice 4.0 or later (its use must be allowed in the SDK settings)
- Apache OpenOffice built-in conversion module (if none of the above are available to the SDK)

To allow and set up the use of the software mentioned above, use the *OfficeConverterSettings* object of the *ImageLoadingParams* object.

To achieve the best quality of data extraction for born-digital documents, the SDK always uses the text layer of such documents obtained by using one of the software products listed above. In the case of imported PDF files with a text layer, the text layer is compared to the text obtained through OCR and if the two texts are identical, then the text layer is used; otherwise, the text obtained through OCR is used. You can also set up the ABBYY FlexiCapture 12 SDK to always use the text layer of PDF documents if you are sure of the quality of their text layer. This is done by modifying the *SourceContentReuseMode* property of the *OfficeConverterSettings* object.

5.1.7 ABBYY FlexiCapture 12 Developer's Package

To simplify the initial setup of the data capture process, the ABBYY FlexiCapture 12 SDK is shipped with an ABBYY FlexiCapture 12 Developer’s Package, which includes FlexiCapture Project Setup Station and FlexiLayout Studio components. The Developer’s Package can be used to set up ABBYY FlexiCapture projects for their subsequent use in the ABBYY FlexiCapture 12 SDK.

All projects created using the Developer’s Package can be used by the SDK and all projects created using the SDK can be used by the Developer’s Package.

5.1.8 Sample Code Library

The ABBYY FlexiCapture 12 SDK distribution package contains sample source code that shows how the SDK can be used in different scenarios. Developers are free to reuse this sample code in their own programs.

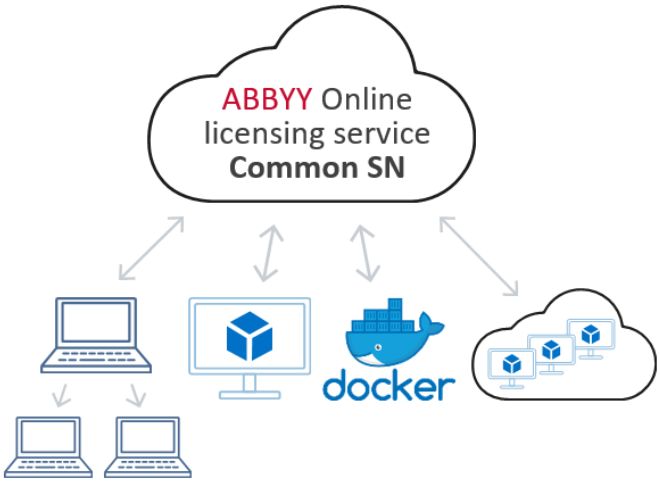
The sample code is provided in C#, C++, Visual Basic .NET, Java, and scripting languages. The distribution package contains the following code samples:

Name	Description
Hello (C++, C#, Visual Basic .NET, Java, JScript, VBScript, Perl)	Shows how to capture data using the ABBYY FlexiCapture 12 SDK with just a few lines of code.
Classification (C++, C#)	Shows how to train and use a document classifier.
Field Extraction Training (C++, C#)	Shows how to train field extraction using feedback from verification operators.

Invoice Processing and Training (C++, C#)	Shows how to create a FlexiCapture project for processing invoices, populate and use a vendor data set, and train field extraction using feedback from verification operators.

5.1.9 Online Licensing

Special online licensing is available for deploying the ABBYY FlexiCapture 12 SDK in containers (including Docker), on virtual machines, and in public cloud environments.



Online licensing requires a permanent Internet connection between the ABBYY Online licensing service and the licensing service running on the customer's machine. The licensing service on the customer's machine will periodically provide a report to the ABBYY Online licensing service containing the number of pages that have been processed on the customer's machine. ABBYY Online licensing service will provide a "work permit" in return. By default, ABBYY Online licensing service and the customer's licensing service will communicate every 5 minutes. No personal information is transferred during this communication.

If the connection between the customer's machine and the ABBYY Online licensing service is interrupted, the ABBYY FlexiCapture 12 SDK will continue to work in autonomous mode for a certain period of time. If, during this time period, the connection is not restored, the ABBYY FlexiCapture 12 SDK will not receive a "work permit" from the ABBYY Online licensing service and the customer will no longer be able to run it.

ABBYY Online licensing service availability is 99% per quarter.

5.2 Release 1 Update 2

5.2.1 Ability to load document definitions into the project

To migrate from FCE to FC SDK, customers may need to switch from FlexiCapture processor to FlexiCapture project because FlexiCapture processor is not supported in FC SDK. In this case they need to create a FlexiCapture project and move all previously used document definitions into it. Previously they could only do it via GUI in FlexiCapture Developer's Package.

Now it is possible by means of API using the new LoadFromFile() method of DocumentDefinitions collection.

5.2.2 Machine learning for processing invoices with different layouts from the same vendor

When one vendor issues invoices with different layouts, training one FlexiLayout for all possible layouts from that vendor will not produce very good results.

The feature for invoices processing allows customers to train several FlexiLayouts for one vendor. Similar-looking images in the training batch should be processed by one layout.

To use this feature in the project, enable clustering feature using the InvoiceFeatures property of the InvoiceSettings object. Find details about invoices processing without data sets in "Recognizing Invoices with ABBYY FlexiCapture SDK" article of the product documentation.

5.2.3 PO numbers, BU and Vendor names extraction from invoices based on neural networks

5.2.3.1 Neural networks used for extracting PO numbers

Earlier, PO numbers could only be extracted using databases or regular expressions. Now PO numbers are extracted using neural networks and such extraction is enabled by default.

To disable PO# extraction based on neural networks add the register key and set a value as described:

[Computer\HKEY_CURRENT_USER\Software\ABBYY\SDK\12\FlexiCapture SDK\DAForms]

EnableNeuralNetPONumber = false

Extraction based on neural networks is supplementary to other methods so it will work in case PO number was not extracted with other methods such as database lookup or search using regular expression.

5.2.3.2 Extraction of BU name and Vendor name based on neural networks

Neural Network is also trained to extract Vendor name and BU name. This extraction will work in both cases: no vendor or BU databases are connected, or databases are connected but database lookup was not successful.

5.2.4 PO number extraction training on user's side

Extraction of PO numbers is now trained on the user's side. For a recurring PO number, only the first instance will be used for training.

For new invoice projects, training of PO number will be enabled by default. When using an invoice project created in an earlier version, please follow these steps:

1. Open your project in Developer's Package. Your project will be updated.
2. Open the *Document Definition Properties* dialog box, click the *Invoice Settings* tab, and make sure that the following options are selected:
    a. Purchase order matching
    b. Thorough extraction of invoice header fields
    c. On the *Event Handler* tab, select *After document state changed*
    d. For the TrainablePO group of fields, select the *Copy PO data to TrainablePO* rule
3. Check the recognition setting of fields from *TrainablePO* group and if they have *Do not recognize* setting on Recognition tab, change it to *Standard Recognition*
4. Start training PO number extraction.

5.2.5 Improved detection of line items with multi-line description

In case of multi-line product or service descriptions, a line item could be separated into several lines. Extraction of such line items has been improved so now the program interpret them like single line item with long description.

Please note that the extraction of multi-line description of the last LineItem on the page may fail.

The mode of extraction of LineItems can be changed using registry key:

[Computer\HKEY_CURRENT_USER\Software\ABBYY\SDK\12\FlexiCapture SDK\DAForms]
"EnableExtendingDescriptionInLineItems" = true

5.2.6 Working with Online licenses using proxy servers

In systems with high security requirements, an Internet connection sometimes are allowed only using a proxy server. In this case, the Online serial numbers become inapplicable - they require a direct connection to the specified addresses and ports.

To meet this requirement, Online licensing now can work using proxy servers. It is possible to setup proxy server credentials using ProxyServer tag in LicensingSettings.xml file. Find details about the settings in "About the LicensingSettings.xml File" article of the product documentation.

5.2.7 Online product documentation

Now it is possible to view the product documentation without installation of the product from any device.

FlexiCapture 12 SDK documentation is published online and available on help.abbyy.com.

Important limitation: access to the documentation is limited and granted to authorized users only. Authorization is organized by Customer Project ID from the FlexiCapture 12 SDK serial number.

5.2.8 Other

5.2.8.1 XLSX export format support

To achieve feature parity with FlexiCapture, now FC SDK supports export XLSX. Previously if there was a setting of export to XLSX in the project, data was exported to XLS.

5.2.8.2 Support of Windows Server 2019

As a product for developers, FlexiCapture SDK support all popular and modern working environments, including the latest available version of MS Windows Server. FC SDK supports MS Windows Server 2019 starting from this product update.

5.2.8.3 Demonstration of FC SDK usage in Python

FC SDK distributive includes a code sample "Hello" in Phyton.

