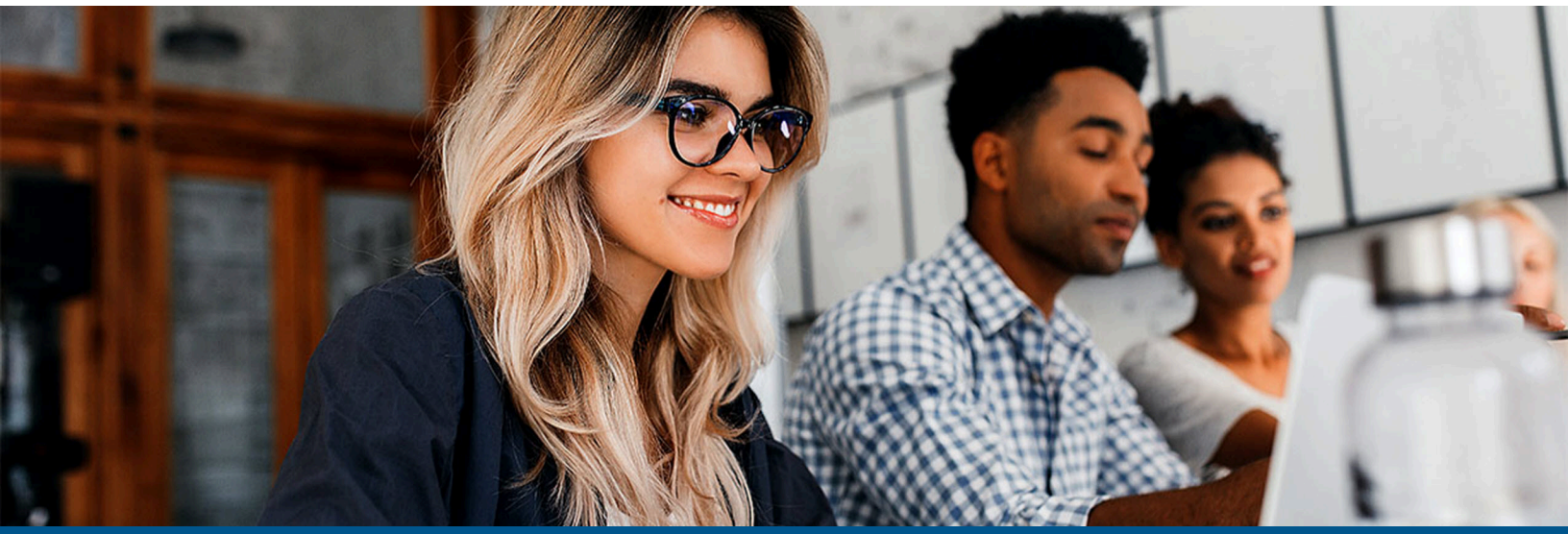


# Kofax Transformation Toolkit
## Developer's Guide

Version: 7.1.0

Date: 2023-04-12

KOFAX

# Table of Contents

# Preface

This guide contains information about installing Kofax Transformation Toolkit and assumes that you have a thorough understanding of Windows standards, applications, and interfaces, as well as Kofax Capture.

This guide is for solution integrators who are installing Kofax Transformation Toolkit or who need a description of the installation procedures and requirements.

Kofax Transformation Toolkit consists of a set of components and additional tools. In this document, the components are also referred to by their module or tool name, omitting the product name.

## Related documentation

The following documentation is available in Kofax Transformation Toolkit 7.1.0.

**Guides**

The following guides are available in Kofax Transformation Toolkit.

***Kofax Transformation Toolkit Developer's Guide***

This guide provides installation information for the Kofax Transformation Toolkit. It also includes details about the provided merge modules and code samples, as well as a detailed information concerning integration.

***Kofax Transformation Toolkit Thin Client Server Developer's Guide***

This guide provides installation and configuration information and provides details about the provided `FileAccessBackend` sample as well as information concerning customization and integration.

***Kofax Transformation Toolkit Kofax Reporting Getting Started Guide***

This guide provides instructions for installing and configuring Kofax Reporting at a client site. In addition, it contains information about the data that is reported and the views that are provided by the reporting database.

**Help**

The following help systems are available.

***Kofax Transformation - Project Builder Help***

This help provides information for creating, configuring, and maintaining a transformation project.

***Kofax Transformation Toolkit Thin Clients Help***

This help includes information on the following Thin Client user modules.

- Thin Client Correction. This section contains information on how to correct extraction problems on a document.
- Thin Client Verification. This section contains information on how to verify pieces of information on a document.
- Thin Client Validation. This section contains information on how to validate problems on a document.

***Kofax Transformation Toolkit Scripting Help***

This help provides scripting examples and how to use the supported script elements.

***Kofax Transformation - XDoc Browser Help***

This help provides information on how to use the XDoc Browser.

***Kofax Transformation - Project Merge Tool Help***

This help provides information on how to merge two versions of the same project.

***Kofax Transformation - Image Classifier Help***

This help provides information about the image classification technology that is used by Statistics Viewer.

***Kofax Transformation - Statistics Viewer Help***

This help provides information about the various statistics that are collected for your solution.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

**Other**

The following additional documentation is available.

***WCFValidationServer.Backend.chm***

This compile help file provides information about the `WCFValidationServer.Backend` library for the Thin Client Server and its `FileAccessBackend` sample.

***Help for Developers***

For Kofax Transformation Toolkit, additional documentation can be displayed that is located in the `<installation path>\Resources\Documentation` folder.

Kofax Transformation Toolkit installs .chm files that contain the following documentation for the Document Review, Correction, Validation, and Validation controls, as well as the Scheduler, NewSamples and Statistic interfaces:

- Kofax.DocumentReview.chm - Kofax.DocumentReview library information
- Kofax.Correction.chm - Kofax.Correction library information
- Kofax.Validation.chm - Kofax.Validation library information
- Kofax.Verification.chm - Kofax.Verification library information
- Kofax.Mailroom.NewSamples.chm - Kofax.NewSamples library information
- Kofax.Mailroom.Statistic.chm - Kofax.Statistic library information

- Kofax.Server.Scheduler.Interface.chm - library information for the server scheduler interface

More information on documentation can be found in Access Kofax Transformation Toolkit documentation.

## Training

Kofax offers both classroom and computer-based training that help you make the most of your Kofax solution. To learn more about training courses and schedules, visit the Kofax Education Portal on the Kofax website.

## Getting help for Kofax products

Kofax regularly updates the Kofax Support site with the latest information about Kofax products.

To access some resources, you must have a valid Support Agreement with an authorized Kofax Reseller/Partner or with Kofax directly.

Go to http://www.kofax.com/support/ for:

- Access to product knowledge bases.
  Click **KNOWLEDGE Base**.
- Access to the Kofax Customer Portal (for eligible customers).
  Click **Kofax Customer Portal** and log in.
  To optimize your use of the portal, go to the Kofax Customer Portal login page and click the link to open the *Guide to the Kofax Support Portal*. The guide describes how to access the portal, what to do before contacting the support team, how to open a new case or view an open case, and what information to collect before opening a case.
- Product information and release news
  Click **Transformation > Kofax Transformation Toolkit**.
- Downloadable product documentation
  Click **Transformation > Kofax Transformation Toolkit > Documentation** and select a document.
- Access to support tools
  Click **Tools** and select the tool to use.
- Information about the support commitment for Kofax products.
  For more information on the support commitment, click **Learn More**.

Chapter 1

# Overview

Kofax Transformation uses advanced recognition technologies to transform scanned images into structured information that can be passed on to back-end systems for further processing.

For an environment where you want a tighter integration of transformation capabilities into existing Windows applications, you can interface directly with the runtime modules using Kofax Transformation Toolkit. The toolkit provides a set of APIs, written in .NET, which enables you to run classification, extraction, and validation from within your own applications.

To use Kofax Transformation Toolkit you must have a working knowledge of Windows programming and the .NET architecture.

The following chapters describe the first steps for running a project, as well as operating requirements, installation procedures, and integration considerations.

Chapter 2

# System requirements

For information on supported operating systems and other system requirements, see the *Kofax Transformation Toolkit Technical Specifications* document on the Kofax Transformation Toolkit Product Documentation site.

This document is updated regularly, and we recommend that you review it carefully to ensure success with Kofax Transformation Toolkit.

## Software prerequisites

The Kofax Transformation Toolkit requires the following software to be installed alongside the product:

These prerequisites are required on production computers and are added as setup programs to the Kofax Transformation Toolkit package so you can include them in your custom installer.

- Runtime libraries for Visual C++ VS2019
- Windows Installer 3.1
- Microsoft .NET Framework 4.8
- Sentinel Protection Installer 7.5.0
- Microsoft .NET Framework 3.5 (prerequisite for Kofax Reporting only)
- Crystal Reports Basic Runtime
  - Runtime libraries for Visual C++ VS2010 SP1 (prerequisite for Crystal Reports only)

Most of these programs are included as part of the Kofax Transformation Toolkit installer, but there is one exception. If you plan to use Kofax Reporting, you need to install the Microsoft .NET Framework 3.5.

In addition to this, if you plan to use the License Server, install this before installing the Kofax Transformation Toolkit.

The license server installer is located in the ISO under `Kofax Transformation Toolkit \Prerequisites\License Server\KofaxLicenseServer-6.5.0.msi`.

Do not change the default installation patch as this causes the Kofax Transformation Toolkit installation to fail.

If required, use the following command via the Command Prompt for a silent installation:

```
msiexec /i KofaxLicenseServer-6.5.0.msi /qn
```

Chapter 3

# Installation

This chapter describes how to install Kofax Transformation Toolkit, and provides an overview of the components that are installed and can be redistributed.

The Kofax Transformation Toolkit package provides a setup for installing applications, as well as a toolkit runtime for integration. The package includes a set of merge modules that you can use to set up your own installation routine and source code examples to show how to use the components for development.

When the installation is complete, the Kofax License Utility starts automatically, enabling you to configure license servers and activate your license.

The Kofax Transformation Toolkit installation provides runtime components and the following design time components:

- Project Builder
- Statistics Viewer

  ⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

- Licensing Server
- Toolkit Design Time Package
- Toolkit Examples
- Toolkit Documentation
- Toolkit Redistributables (merge modules)
- Add-ons (additional OCR engines)
- Toolkit Scheduler Service (including the source code samples)
- Reporting components

The Kofax Transformation Toolkit package also provides additional installers for:

- Kofax Search and Matching Server
- Kofax Transformation Toolkit - Thin Client Server

ⓘ The Kofax Transformation Toolkit cannot be installed on the same server as either Kofax Capture or Kofax Transformation Modules.

# Important installation notes

This section lists important notes for installing Kofax Transformation Toolkit

**Administrator Rights**

To install the product, the logged-in user must have Administrator rights on the local computer.

**Microsoft .NET Framework 4.8**

Microsoft .NET Framework 4.8 is a prerequisite for Kofax Transformation Toolkit.

**Kofax Transformation Toolkit and Kofax Search and Matching Server**

When you upgrade Kofax Transformation Toolkit to the new version and you want to use remote fuzzy databases you also have to upgrade Kofax Search and Matching Server to version 6.8 or newer. If you do not synchronize the Kofax Transformation Toolkit and Kofax Search and Matching Server versions any search requests on the remote fuzzy databases will fail.

**Help Settings**

In order to display the Kofax help correctly you need to ensure that your browser has scripting enabled. Otherwise, the table of contents and the search and index tabs may not be displayed correctly and context-sensitivity does not work.

For a customized installer you need to ensure that the help is copied to a separate folder and that this path is configured in the registry.

# Install Kofax Transformation Toolkit

Kofax Transformation Toolkit must be installed on each workstation where it is used.

If you upgrade from Kofax Transformation Toolkit 7.0 or earlier, you need to remove the earlier version before installing Kofax Transformation Toolkit.

If you want to make use of the parallelization of extraction processes, you have to select the Server Scheduler Service during setup. This feature is deactivated by default.

1. Start the installation by running setup.exe from the root of the folder containing the product files.
2. Follow the instructions presented to you.
3. From the custom setup select optional features you want to install.
    a. Optionally, select "Toolkit Redistributables" to install the merge modules in order to build a custom setup.
    b. Optionally, select additional OCR engine from Add-ons to install. Note that those may require additional licensing.
    c. Optionally, select "Toolkit Scheduler Service" feature to install the Server Scheduler Service.

d. Optionally, select "Examples Using Scheduler Service" feature to install the document and batch processing source code samples that use the Server Scheduler Service parallelization.

e. Optionally, select "Reporting" to install additional components needed to enable Kofax Reporting.

A window is displayed that allows setting the URL to the WSA Receiver.

Type the URL, for example, "http://WSAreceicer:25480" and click **OK**. If needed, you can change the settings later using the Kofax Reporting Client Configuration tool by selecting "Reporting Configuration" from the Kofax Transformation Toolkit start menu folder. Alternatively, you can start the WSA configuration tool from the Kofax Reporting folder. For more details about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

4. When the installation is finished click **Finish** to close the set up.

Repeat the installation on each workstation where you want to run Kofax Transformation Toolkit.

# Folder and registry permissions

To run Kofax Transformation Toolkit, various user permissions for folders and registry keys are required to run various applications.

- Name: **KSALicenseService.exe**
- Description: Licensing for Kofax products
- Default Path: C:\Program Files\Common Files\Kofax\Licensing\Server\KSALicenseService.exe

**KSALicenseService.exe User Permission Requirements**

| Path/Registry Key | Permission |
|---|---|
| C:\Program Files\Common Files\Kofax\Licensing\ and all child directories | Read/Write |
| C:\ProgramData\Kofax\KSALic\logs\ | Read/Write |
| C:\ProgramData\Kofax\AppLogging\DB\ | Read/Write |
| C:\ProgramData\Kofax Image Products\Local\Scripts\ | Read |
| HKEY_LOCAL_MACHINE\ System\CurrentControlSet\Services\EventLog \Application\KOFAX-SAL | Full Control |
| HKEY_LOCAL_MACHINE\ System\CurrentControlSet\Services\Sentinel\Current | Full Control |
| HKEY_LOCAL_MACHINE\ System\CurrentControlSet\Services\EventLog \Application\KSALicenseService | Full Control |
| HKEY_CLASSES_ROOT\ \Installer\Assemblies\C:\|Program Files\|Common Files\| Kofax\|Licensing\|Server\|KSALicenseService.exe | Full Control |
| HKEY_USERS\.DEFAULT\Software\Kofax\SALic | Full Control |

- Name: **KSALicenseUtility.exe**
- Description: Licensing Utility for Kofax products that can be started by a user

- Default Path: C:\Program Files\Common Files\Kofax\Licensing\Server\ KSALicenseUtility.exe

**KSALicenseUtility.exe User Permission Requirements**

| Path | Permission |
|---|---|
| C:\Program Files\Common Files\Kofax\Licensing\ | Read/Write |
| C:\ProgramData\Kofax\KTT\Config\ | Read |
| C:\ProgramData\Kofax\Vrs\ | Read/Write |
| HKEY_CLASSES_ROOT\ \Installer\Assemblies | Full Control |
| HKEY_LOCAL_MACHINE\Software\Kofax\SALicClient | Full Control |
| HKEY_LOCAL_MACHINE\Software\Kofax\SALicClient\SharedLicenseServerFileUnc | Full Control |
| HKEY_CURRENT_USER\Software\Kofax\SALicUtility | Full Control |
| HKEY_CURRENT_USER\ Software\Microsoft\Installer\Assemblies\C:\|Program Files\|Common Files\|Kofax\|Licensing\|KSALicenseUtility.exe | Full Control |

# Use quiet mode installation

If you do not want to perform an interactive installation of Kofax Transformation Toolkit by using the executable, you can use Quiet Mode via the Command Prompt. This type of installation does not display configuration, progress, feature information, or warning windows, although error messages are still displayed. This is ideal if you are installing user modules or specific components across a network because you do not have to visit each machine individually.

You can use Quiet Mode installation to automate the installation of the user interactive modules (Validation, Correction, Document Review, and Verification) on target systems.

By default, the complete Kofax Transformation Toolkit functionality is installed by running the setup executable with the Quiet Mode parameter at the command prompt. For more information an installing or excluding various components from the install, there are a list of supported parameters that you can add to the installation command.

If any of the prerequisites are installed during quiet installation, it is necessary to reboot your system after the installation has finished in order for Kofax Transformation Toolkit to function correctly.

You can install Kofax Transformation Toolkit using the command prompt by following these steps:

1. From the **Start** menu, select **Run**.
2. Type **cmd** and press Enter.
   The Command Prompt window is displayed.
3. Change to the folder that contains the Kofax Transformation Toolkit Setup.exe file.

   If the setup.exe file is located on a network drive, remember to map to the network drive first.

4. Run setup from the command line, using the relevant parameters.

## Parameters

The following parameters are available for the silent installation.

**Parameters for Silent Installation**

| Setup.exe Parameters | Description and Value Settings |
|---|---|
| `/?` | Shows the help window. |
| `/quiet` | Runs the installer with no user interaction. |
| `/log [filename]` | Writes the installer log messages to the specified file. |
| `/uninstall` or `/u` | Uninstalls the product. |
| `/install` or `/i` | Installs the product. |

The following values are available for the property parameter.

**Values for Property Parameter**

| | |
|---|---|
| `NOKSALICENSINGSERVER=1` | Disables License Utility service installation. |
| `STATISTICS=1` | Enables Kofax Transformation - Statistics Viewer installation.<br>⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit. |
| `NORUNTIME=1` | Disables installation of **all** runtime components for integration in Kofax Transformation Toolkit. |
| `NODEV=1` | Disables Kofax Transformation Toolkit Design Time Package installation. |
| `NOEXAMPLES=1` | Disables Kofax Transformation Toolkit examples installation. |
| `NODEVDOC=1` | Disables the installation of the Kofax Transformation Toolkit documentation. |
| `REDIST=1` | Enables Kofax Transformation Toolkit Redistributables installation. |
| `ALLADDONS=1` | Selects all available Add-Ons for installation. |
| `SCHEDULERSERVICE=1` | Enables the Server Scheduler Service installation. |
| `EXAMPLESERVICE=1` | Enables the installation of the Extraction WebService application example. |

| | |
|---|---|
| `REPORTINGURL` | Enables the installation of the reporting configuration tool and configures the URL to the WSA Receiver, which gathers statistical data for the reporting platform. For this parameter you need to set the URL to the receiver and the port number, for example, REPORTINGURL="http://localhost:25481".<br>Use this parameter together with the `REPORTING` parameter; otherwise no statistical can be gathered. |
| `REPORTING=1` | Enables the installation of the Kofax Reporting functionality.<br>Use this parameter together with the `REPORTINGURL` parameter; otherwise gathered statistical data cannot be reported to the reporting platform. |

# Install the Natural Language Processing engine language packs

Before you can use either the Named Entity Locator or the Sentiment Locator it is necessary to install the Natural Language Processing engine language packs. The Natural Language Processing engine performs natural language processing and is able to extract items such as named entities and sentiments, but it requires at least one or more language packs to be installed.

For more information on the Natural Language Processing engine and its supported locators, refer to the *Kofax Transformation Toolkit - Project Builder Help*.

Download the Natural Language Processing engine language packs from the Kofax Delivery site.

You can install the Natural Language Processing engine language packs by following these steps:

1. Extract the downloaded zip file.
2. Navigate to the `MicroPackage` folder in the extracted files and double-click on one of the following `.MSP` files, depending on what languages you are supporting.
   - **KofaxTransformation_SalienceV6.4.0_LanguageBundle_western-default**.
     Run to install English, Spanish, Portuguese, French, and German support.
   - **KofaxTransformation_SalienceV6.4.0_LanguageBundle_western-extended**.
     Run to install Italian, Romanian, and Dutch support.
   - **KofaxTransformation_SalienceV6.4.0_LanguageBundle_extended**.
     Run this to install Japanese, Chinese, and Korean support.

   A Windows Installer window is displayed and then installs the selected Kofax NLP (Natural Language Processing engine) languages automatically.

   The installer window closes when the installation is complete.
3. Optionally, double-click on another installer if you want to support additional languages.

# Uninstall Kofax Transformation Toolkit

You can uninstall an earlier version by following these steps:

1. Shut down any Kofax Transformation Toolkit applications.
2. Start the uninstall by running **Setup.exe** from the Kofax Transformation Toolkit directory of the installation media. Alternatively, you can uninstall Kofax Transformation Toolkit from the Control Panel using **Add and Remove**.
3. In the Kofax Transformation Toolkit Setup Wizard select **Remove**.
4. When Kofax Transformation Toolkit has successfully been uninstalled, click **Close**.

Chapter 4

# Use Kofax Transformation Toolkit

This section introduces the classes available through the Kofax Transformation Toolkit and an description of the concepts involved in processing documents.

## First steps to run a project

After installation of Kofax Transformation Toolkit you need to use a development environment which supports .NET, such as VB.NET or C#. You need to reference the Kofax Transformation Toolkit components in your programming environment. The toolkit and its examples can only be used if a valid hardware key or software license is activated using the Kofax License Utility.

Kofax Transformation Toolkit only provides interfaces to the runtime components. You still need to set up a project using Project Builder, which is installed with the Kofax Transformation Toolkit to define the needed configuration to run recognition, classification and document separation, extraction and validation steps.

Any change to a project through the application interface (API) can lead to invalid settings and is not supported. Best practice is to use the Project Builder to set up a project and configure the needed processing steps.

Once a project is created it can easily be loaded and executed using the Kofax Transformation Toolkit runtime components. The basic components that are available are:

- **CscProject** – objects containing all settings and scripts as defined in Kofax Transformation - Project Builder.
- **CscXDocInfo** and **CscXFolder**/**CscXDocument** – objects representing a batch or a document containing all runtime data generated in the process.
- **Document Review** – one Visual control (ActiveX/.NET) for displaying and enabling the editing of the batch and document structure, including a control to display the current problem.
- **Correction** – one Visual control (ActiveX/.NET) for displaying the fields in the XDocument and enabling the editing of those fields.
- **Validation** – one Visual control (ActiveX/.NET) as an editor for XDocuments and one Visual control (ActiveX/.NET) for displaying the batch structure.
- **Verification** – one Visual control (ActiveX/.NET) as an editor for XDocuments. For displaying the batch structure you can use the Visual control from the Validation component.

These are the simplified steps to process an image:

- First create an XDocument and import an image, text, or PDF.
- Load an existing Project from file.

- Call the classify method of the Project passing the XDocument reference as a parameter.
- Call a Page Recognizer to perform OCR as needed (event driven).
- Call the extract method of the project passing the XDocument reference as a parameter.
- The XDocument by then contains all the result of OCR, classification and extraction. Save the XDocument or save the results of the process into your back end system.

Once an image is processed and stored as an XDoc it can optionally be displayed, validated and edited using the Validation control. These are the simplified steps to display and validate an XDocument:

- Load the Project from file that is used to create the XDocument and initialize the Validation control with it. The Project contains all validation settings that were configured using Project Builder, they are applied to the XDocument by the Validation control.
- Load the XDocument from file and pass it to the Validation control.
- Save the changed XDocument to disk.

# Class overview

The following section contains a list of classes available in the Kofax Transformation Toolkit. These classes are referenced in the sample projects provided with the installation.

The Kofax Transformation Toolkit is composed of several COM components and ActiveX controls. All these can be used in various combinations to build "document transformation" applications.

The most important classes/components are as follows:

- Project
- XDocument
- XFolder and XDocInfo
- Licensing
- Validation
- Online Learning
- Verification
- Statistics
- Correction
- Document Review

## Project

**Class name**: CscProject

**Component**: Kofax Cascade Project 2.0

This is a class which contains all settings and configurations created in Kofax Transformation - Project Builder.

These include:

- OCR profiles used for page or zone recognition and their settings

- Class names, class hierarchy and the classification settings
- Locator definitions to extract the document information
- Field definitions at class level
- Formatting methods
- Validation methods
- Validation form layouts
- Training and sample documents
- Scripts

A project can be used to classify, extract, and validate an XDocument (XDoc). This means that the XDocument can be populated with the classification and extraction results. A project is saved and loaded from an .fpr file.

## XDocument

**Class name**: CscXDocument

**Component**: Kofax Cascade XDoc 2.0

This is a class that represents a document and all data associated with it during the process of classification, extraction and validation.

This class can contain, but is not limited to the following:

- A reference to the original source images or text files
- The full page OCR data, organized as collections of pages. These pages are composed of collections of text lines and collections of words
- The classification result for any classification performed on a document. The fields assigned to the class are created as soon as the classification result is assigned to an XDocument
- A list of all results and possible alternatives the locator found on that document during extraction. The field results which are retrieved from the locator results according to the specified field settings.

## XFolder and XDocInfo

**Class name**: CscXFolder, CscXDocInfo

**Component**: Kofax Cascade XDoc 2.0

The CscXFolder class represents a batch and its structure and contains basic information about the contained documents such as document page count and the image file names. As this information is part of the CscXDocInfo document wrapper so that the documents themselves do not need to be loaded.

The XFolder class contains the following:

- Collections of all XFolders and XDocInfos in that XFolder
- Folder fields
- Internal collection of keyed XValues that can be used to attach specific information.

The XDocInfo class contains the following information:

- XDocument (this property contains a reference to the XDocument)
- Access to properties of the XDocument without loading the XDocument like filename, image file names, page count and valid state
- Internal collection of keyed XValues used to attach specific information (these XValues are populated to the XDocument object during load)

The root XFolder object can be saved in a file and loaded from this file. The XFolder object is used in combination with Foldering and Batch Editing in the Kofax Transformation Toolkit components.

## Licensing

The licensing classes exist for the core Kofax Transformation Toolkit components. This includes:

- CscXDocLicensing
- CscProjectLicensing
- CscImageLicensing

These license classes must to be initialized in conjunction with an activated valid hardware key or a software license before the Kofax Transformation Toolkit functionality is available. This has to be done for each Kofax Transformation Toolkit application. The source code samples show how to use the provided Kofax Transformation Toolkit functionality, for example, the BatchServer source code sample.

For a clean implementation it is best practice to disconnect from the licensing server after a module is terminated.

The licensing objects reference the hardware or software license that is activated using the Kofax License Utility.

### Kofax License Utility

Kofax Transformation Toolkit licensing is managed through the Kofax License Utility. The tool is displayed immediately following the installation of the toolkit. The Kofax License Utility enables you to perform the following:

- Configure the primary and optional backup license servers
- Activate a software license
- Activate a hardware license
- View the active licenses for your installation

The Kofax License Utility application is installed with the full setup. The installation destination is:

```
…\Program Files\Common Files\Kofax\Licensing\KSALicenseUtility.exe
```

The application allows you to configure and view the licenses and volume counters on the hardware key or software license.

For information regarding how to configure the Kofax License Utility, please refer to the *Kofax License Utility Help* .

## Classification Online Learning

Classification Online Learning has two distinct modes. The first mode collects documents only while the other mode creates dynamic classifiers that are used by the Server. If you want to support Classification Online Learning with dynamic classifiers, ensure that the data needed to feed the classifiers is available in the online learning directory.

This has to happen in your back end administration application where the project is 'published'. You need to call the following code lines to ensure that the data is available in the online learning directory.

```
If project.ClassificationOnlineLearning.IsActiveInServer() Then
    ClassificationOnlineLearningPublisher.Publish(project)
End If
```

In order to execute the above code, you need access to the `Kofax.NewSamples.dll` file.

## Extraction Online Learning

**Component**: Kofax.Mailroom.NewSamples

This class provides Online Learning functionality by collecting sample documents marked in validation and processing documents for Generic, respectively Specific Online Learning. For Specific Online Learning the Online Learning Manager library is used to create and train dynamic Knowledge Bases which can be used for the next document processing (Server processing), if wanted; and additionally stores documents allowing later import into Project Builder to improve a project for better extraction results. For Generic Online Learning this "Knowledge Base Learning Server" process stores documents in a database, which can be displayed in the Project Builder to improve a project for classification, extraction or table recognition.

For further details concerning the concept of Online Learning, see the *Kofax Transformation - Project Builder Help*.

## Validation

**Class name**: Kofax.Validation.COM.AxInterop.AxValidation

**Component**: AxKofax.Validation.COM / Kofax.Validation.COM

This is a control for the user interface and functionality offered in Kofax Transformation Toolkit - Validation. The Validation control should be placed on a Windows form. During processing it is initialized by a `Project` object. The XDocument requiring validation is then passed to that control. On the basis of the XDocument the validation control is initialized to display the document data. To do this, it uses the definition data from the `Project` object (for example, the validation form layout specified for the class). The validation control applies the defined validation rules and formatting methods to the entered data during user interaction. The Validation control allows reclassification, interactive viewing, validation and editing of any extracted fields from a document.

After modification of an XDocument in the Validation control the document can be saved containing all changes.

**Class name**: Kofax.Validation.COM.AxInterop.AxBatchView

**Component**: AxKofax.Validation.COM / Kofax.Validation.COM

This "tree-view" control is used to display a batch structure. It allows an interactive editing mode to manipulate documents within a batch structure. For all actions events are provided to transmit the changes on the fly to the integrating system.

It provides several navigation methods and fires events for all changes of selection.

**Class name**: Kofax.Validation.COM.AxInterop.AxDocViewer

**Component**: AxKofax.Validation.COM / Kofax.Validation.COM

This control is used to display document images during Batch Editing. When Batch Editing is selected the Validation control is hidden and the document displays in this simple Document Viewer.

## Verification

**Class name**: Kofax.Verification.COM.AxInterop.AxVerification

**Component**: AxKofax.Verification.COM / Kofax.Verification.COM

This is a control for the user interface and functionality offered in Kofax Transformation Toolkit - Verification. The Verification control should be placed on a Windows form. During processing it is initialized by a `Project` object. The XDocument requiring verification is then passed to that control. On the basis of the XDocument the Verification control is initialized to display the document data. To do this, it uses the definition data from the `Project` object. This can be defined in Project Builder by configuring the field for verification in the Field Details User Interactive Modules group.

After modification of an XDocument in the Verification control the document can be saved containing all changes.

For displaying the batch structure you can use the "tree-view" control from the Validation library. Batch editing actions are not allowed for the verification step.

## Statistics

**Component**: Kofax.Mailroom.Statistics

This class is used to collect runtime statistical data regarding extraction and validation. These statistics can then be viewed using Statistics Viewer.

The Statistics Viewer interface supports data aggregation, which means that detailed data is aggregated depending on a defined cleanup interval. Each time this interval passes, existing single data sets are added to a summarized data set.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

## Correction

**Class name**: AxKofaxCorrectionCOMLib.AxCorrection

**Component**: Kofax.Correction.COM.Interop
Kofax.Correction.COM.AxInterop
AxCorrection

This is a control that contains a user interface and functionality equivalent to that in Kofax Transformation Toolkit - Correction. The Correction control is placed on a Windows form. During processing it is initialized by a Project object and the XFolder representing the batch.

The Correction control reads the definition data from the Project object (that is, the fields to be displayed in correction and their order) for displaying the data.

During user interaction, the Correction control applies the defined single field validation rules to the entered data. The Correction control acts as an editor of the XDocument which allows interactive viewing, correction and editing of configured fields from a document. After modification of an XDocument in the Correction control, the document can be saved containing all changes.

### Document Review

**Class name**: AxKofaxDocumentReviewCOMLib.AxDocumentReview

**Component**: Kofax.DocumentReview.AxInterop
Kofax.DocumentReview.Interop
AxDocumentReview

This is a control that contains a user interface and functionality equivalent to that in Kofax Transformation Toolkit - Document Review. The Document Review control is placed on a Windows form. During processing it is initialized by a Project object and the XFolder representing the batch.

The Document Review control displays the structure of the XFolder and problems so users are able to assign classes or confirm suggested classes for unclassified documents, split or merge documents, move pages between documents, and reorder pages within documents. After modification of a document in the Document Review control, the corresponding XDocument and parent XFolder are updated automatically. These changes can then be saved to an XDocument on disk.

The batch based samples are the only samples that support JPEG and PNG images. This sample only supports *.tif, *.txt, and *.pdf file formats.

## Process documents

In general there are two concepts of processing documents by the Kofax Transformation Toolkit. One concept is to deal only with single documents. Then from the document source an XDocument is created that is used for server processing, the validation, the verification, the knowledge base learning server and the statistics.

The other concept is based on batches. The configured input directory is used by the server application to create a batch structure that is saved to a file. That `XFolder` object which represents the batch is then used by all following applications to process the documents. The batch concept is mandatory when foldering is used or the batch view control in the validation application is applied.

The XFolder object gives access to the XDocument objects through a document wrapper object, called XDocInfo. These `XDocInfo` objects contain the `XDocument` objects that are used in the single document concept directly.

In the following sections the basic workflow for single document processing is explained. Since the concept of single document processing is almost similar to the one of the batch processing, the section about the batch processing concept only explains the differences.

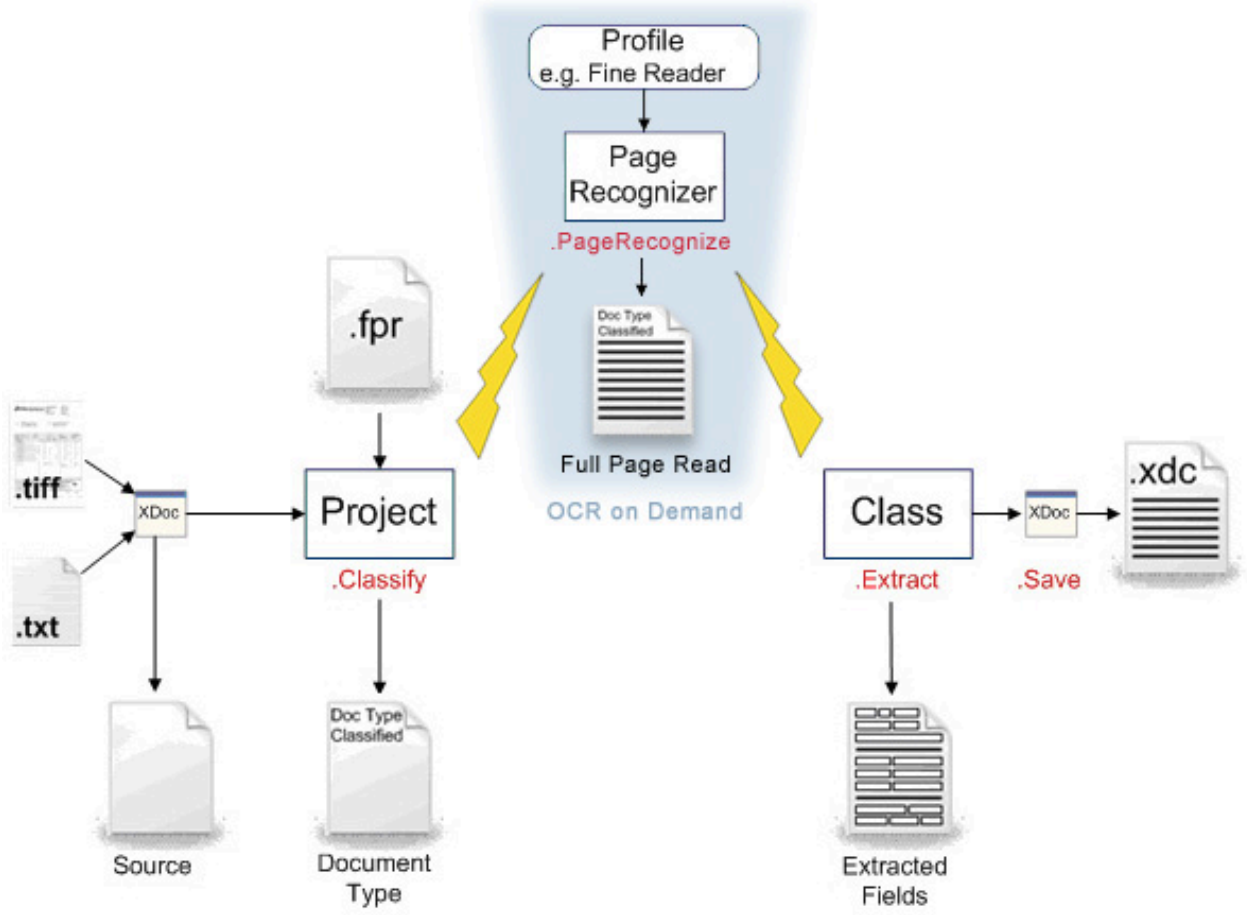
## Process single documents

The following sections show examples of what steps are needed to create a basic workflow with the Kofax Transformation Toolkit when single documents are used for processing.

### Classify and extract a single document

A document is processed through a custom workflow. A source file (*.tif, *.txt, or *.pdf) is imported into an `XDocument` object (represented as 'XDoc' in the example below). That XDocument is then passed to the `Project` object which classifies the document based on the settings in the `Project` object and writes that information to the `XDocument` object. OCR is executed on demand that means if OCR is needed for the classification step, because text or instruction classification is used, OCR is executed. The OCR is executed by passing the `XDocument` object to a page recognizer that performs OCR corresponding to its settings and updates the XDocument file with that gathered information. The next step is the extraction of the XDocument. Here again if the OCR is not executed yet, it is executed if needed for the defined extraction methods.

Next the document is extracted, the defined formatting and validation steps are performed and the results are written to the XDocument object. For more information about the steps performed during extraction see *Project Builder Help - Setup Validation - Validation Sequence*. The final `XDocument` object contains all gathered data and can be saved to disk. This file can be forwarded to the next step in the workflow.

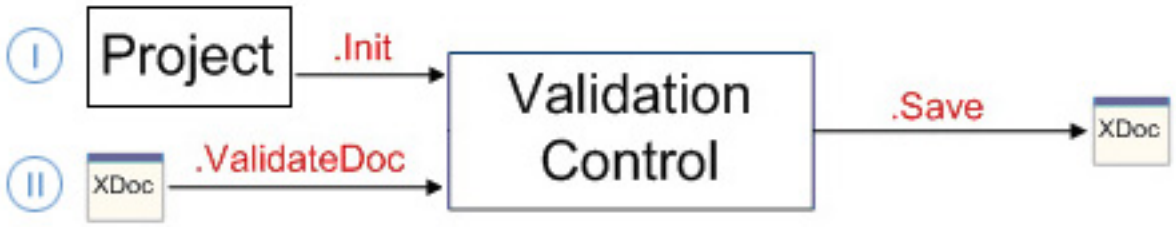



For further information see the code sample DocumentServer.

## Validate a document

In order to validate an XDocument using Kofax Transformation Toolkit, the validation control must be initialized with the project file that is used for the document processing. This processed XDocument must be loaded and passed to the Validation control component.

The validation rules defined in the project file are applied during the manual validation of the document in the validation control. All changes applied to the XDocument during validation have to be saved to disk.




For further information see the code sample ValidateDocument.

## Process document for online learning

A document can be marked for Specific or Generic Online Learning in the validation step.

If a document is flagged for Specific Online Learning the following OnlineLearningManager step stores it and creates specific Knowledge Bases which can be used for the next document processing to improve extraction results (as long as the corresponding setting is selected in the project settings for a project.) In a separate step the trained documents can be imported to the Project Builder and used to modify the project file to improve extraction results. For further information about **Specific Online Learning** please see the *Kofax Transformation - Project Builder Help* and refer to the section "Invoice Processing Technology - Concept of **Specific Online Learning**".

If a document is flagged for **Generic Online Learning** the following Knowledge Base Learning step saves it to a database. Later the project administrator decides how to handle the gathered documents, for example, to train the project for better classification and extraction results, or to add a new class to improve table extraction.


For further information, see the Knowledge Base Learning Server code sample.

## Verify a document

In order to verify an XDocument using Kofax Transformation Toolkit, the verification control must be initialized with the project file that was used for the document processing. The processed XDocument must be loaded and passed to the Verification control component.

The verification rules defined in the project file are applied during the manual verification of the document in the verification control. All changes applied to the XDocument during verification have to be saved to disk.


For further information see the code sample VerifyDocument.

## Gather document statistics

To collect runtime statistics an XDocument needs to be loaded and then its data written to a statistics database. This gathered information can be analyzed with the Statistics Viewer application. This application contains a couple of predefined reports.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

For statistical analysis the document information must be retrieved during server processing (please refer to the server samples).

If no database is found a new database is created. For starting with a batch a data set has to be created. Then for each document the relevant information is extracted and written to the database.

The Statistics interface supports data aggregation, means that detailed data is aggregated depending on a defined cleanup interval, e.g. single data sets are added to a summarized data set.

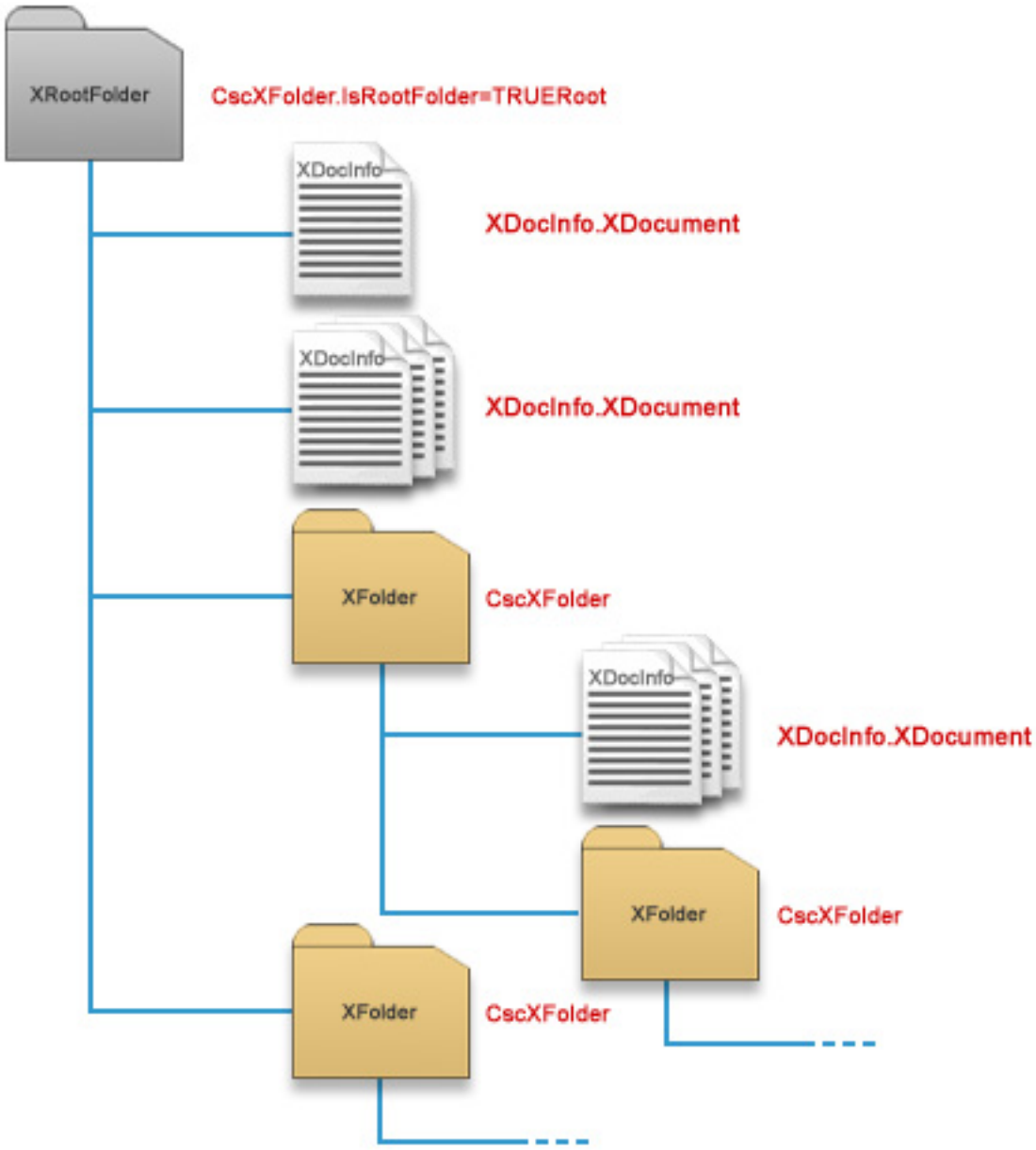

For further information, see the Statistics server code sample.

## Concept of batch processing

The following sections show examples of what steps are needed to create a basic workflow with the Kofax Transformation Toolkit when batches are used for processing.

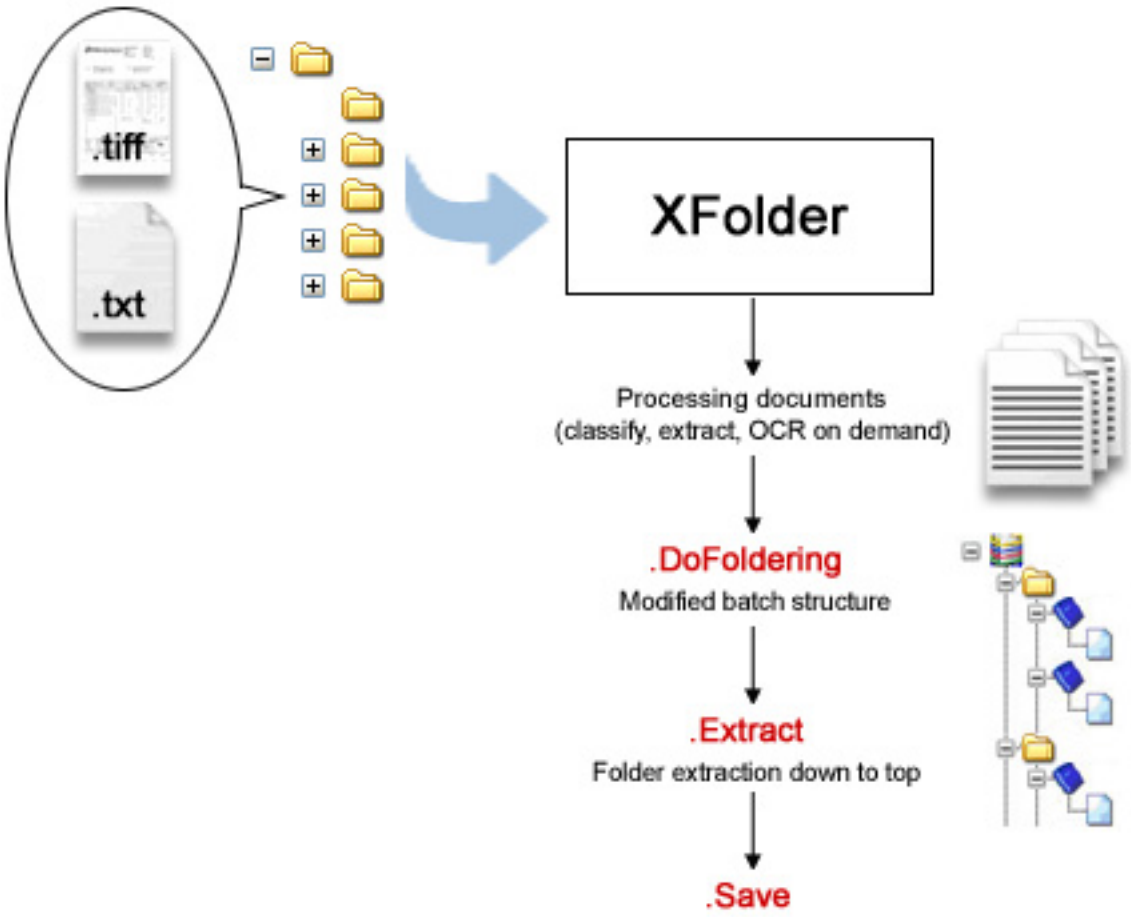
The server application creates a batch structure that is saved to a file. The XFolder object that represents the batch must be used by all following workflow steps to process the documents. The batch concept is mandatory when foldering is used or the batch view control is used in the validation application.

The XFolder object gives access to the XDocument objects through a document wrapper object called XDocInfo. These XDocInfo objects contain the XDocument object which is used in the single document concept.




Source files for all supported file formats in a directory are imported to a XFolder object. This import can be done recursively (considering existing subdirectories and reflecting them in the batch structure) or flat. This XFolder object then is used to process first all documents with the same processing steps as described in the single document processing (classify, extract, OCR on demand).

After document processing the foldering for the XFolder object is executed. During this step the structure of the batch can be modified. Then the extraction for all subfolders is performed in a depth-first manner. In this step modifications of the documents can happen (changes in fields implemented in script) so the documents have to be saved again.



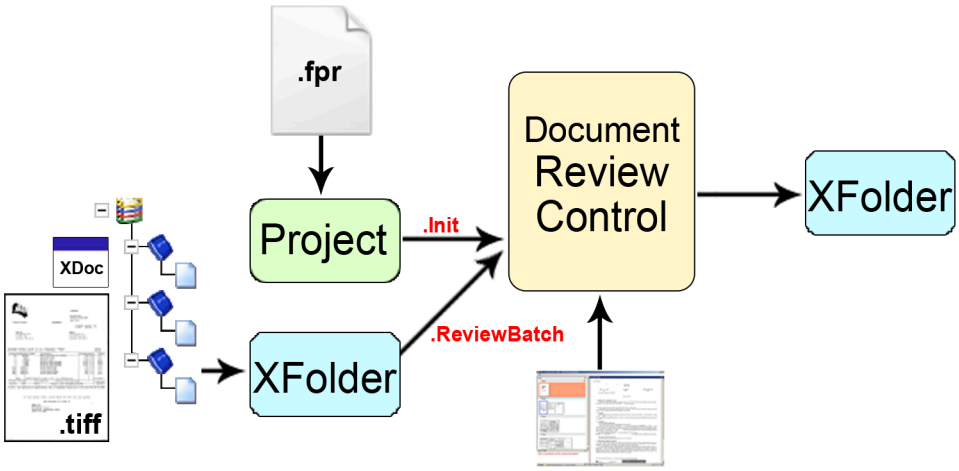

For further information, see the Batch server code sample.

## Review a batch

In order to review a batch using the Document Review control, it must first be initialized with the project file that is used to classify each document during processing. This project file also contains all of the Document Review rules configured in Project Builder that are needed to review the batch. The rules are used by the Document Review control to identify problems in the batch which require the attention of the user.

Once the Document Review control is initialized with the appropriate project file, the XFolder representing the currently processed batch needs to be passed to the control. The XFolder then is displayed with any problems highlighted. The Document Review rules are then re-run on the batch each time a change is made.

It is possible to zoom in and zoom out using the FullImageZoomIn and FullImageZoomOut methods.




For further information see the code sample BatchDocumentReview.

## Correct a batch

In order to correct a batch using the Correction control from the Kofax Transformation Toolkit, the Correction control must be initialized with the project file that is used for the processing of the batch and the XFolder that represents the processed batch. The Init function is used to initialize the Correction control.

Fields on the Correction control may be displayed in order of field name, or in order of the document they belong to (and then by field name). Selection of the field is set with the `DataFlow` property.

Single field validation rules defined in the project file is applied to the field automatically as the field is corrected. In case the validation rule fails, the cause of failure is provided via the `ValidationException` property.

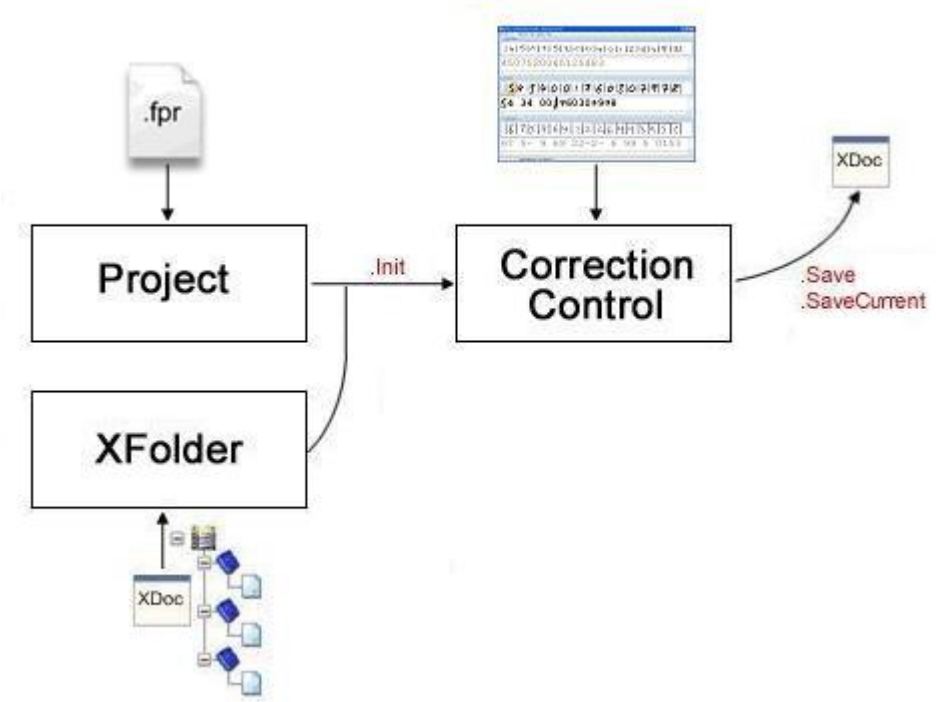
Changes made to the field may be saved in three ways:

- If the `FastSave` property of the control is set to true, then all changes made to the field are saved into the Xdocument automatically as the field is corrected.
- The user may use the `SaveCurrent` function to manually save the data of the field being processed. This may be used in the `FieldCorrected` event.
- The `Save` function may be used for saving corrected data. The Save function saves all the corrected fields in the batch. This function may be used at the end of processing of the batch in the `BatchElaborated` event handler.

When the Correction control is switched to the Full Image mode the control shows only the image with the zone drawn around the currently corrected field. It is possible to zoom in and zoom out using the `FullImageZoomIn` and `FullImageZoomOut` methods.

The Correction control has the setting to copy the value from the previously corrected field to the field which is currently being corrected. This functionality is called AutoDupe, and can be turned on by setting the AutoDupe property to true. To copy the value from the previous field into current one, the PerformAutoDupe method must be called.

The Correction control has the setting to show the value recognized by the extraction in the tool tip displayed on the image snippet of the field. This functionality may be turned on by setting the showOCR property to true. The OCR result may be set as the field value by calling the AcceptOCRData method.




For further information see the code sample BatchCorrection.

If you have Thin Client Server installed, you can also use Thin Client Correction to validate a batch. Note that you have to enable thin clients and especially the Enable Thin Client Validation setting in the project settings in order to process a batch with Thin Client Correction. For further information, see *Kofax Transformation Toolkit - Thin Client Server Developer's Guide* .

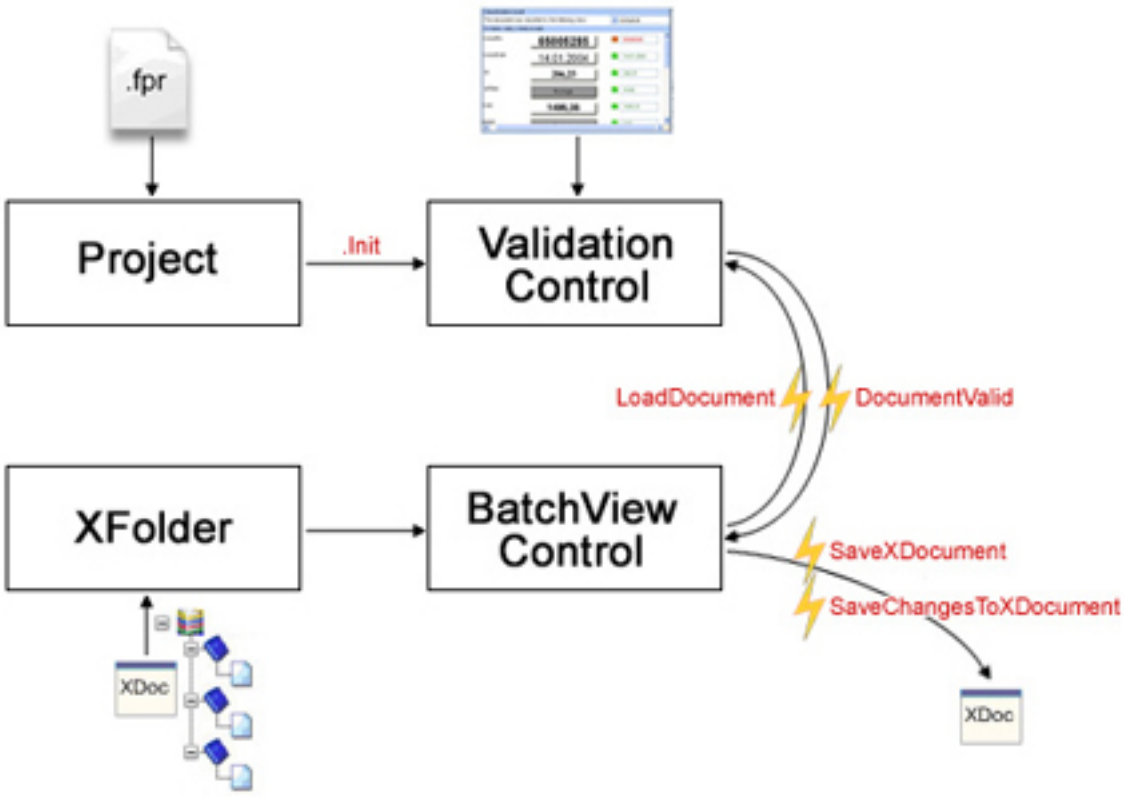
## Validate a batch

In order to validate a batch using the Validation and the BatchView controls from the Kofax Transformation Toolkit, the Validation control must be initialized with the project file that is used for the processing of the document and the BatchView control must be initialized with the loaded XFolder object which represents the processed batch. The LoadDocument event of the BatchView control is used to initialize the Validation control with the document to validate.

The validation rules defined in the project file are applied during the manual validation of the document in the Validation control. The changed XDocument has to be saved to disk in

the SaveXDocument and the SaveChangesToXDocument event of the BatchView control. The SaveChangesToXDocument event allows the possibility of canceling the saving of changes.

If the batch editing functionality of the BatchView control is used an additional DocumentViewer control is needed that temporarily substituted for the Validation control to avoid user interaction on the document during Batch Edit.




For further information see the code sample BatchValidation.

If you have Thin Client Server installed, you can also use Thin Client Validation to validate a batch. Note that you have to enable thin clients and especially the Enable Thin Client Validation setting in the project settings in order to process a batch with Thin Client Validation. For further information, see *Kofax Transformation Toolkit - Thin Client Server Developer's Guide* .

## Process a batch for online learning

Processing a batch for online learning applies the same concept as described in the section Process Document For Online Learning. The only difference is that the documents are retrieved from the loaded XFolder object. The folder structure itself is not considered for online learning.

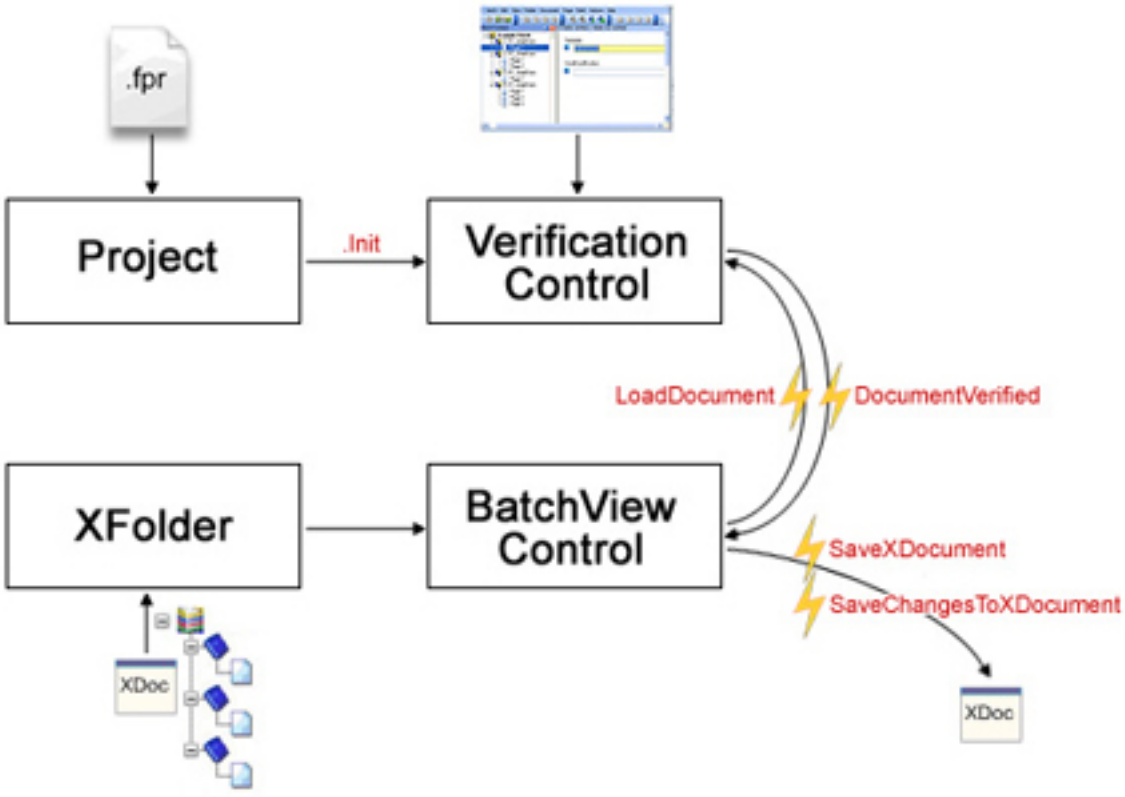
For further information see the code sample KnowledgeBaseLearningServer.

## Verify a batch

In order to verify a batch using the Verification and the BatchView controls from the Kofax Transformation Toolkit, the Verification control must be initialized with the project file that is used for the processing of the document and the BatchView control must be initialized with the loaded XFolder object which represents the processed batch. The LoadDocument event of the BatchView control is used to initialize the Verification control with the document to verify.

The verification rules defined in the project file are applied during the manual verification of the document in the Verification control. The changed XDocument has to be saved to disk in the SaveXDocument and the SaveChangesToXDocument event of the BatchView control. The SaveChangesToXDocument event allows the possibility of canceling the saving of changes.

If the batch editing functionality of the BatchView control is used an additional DocumentViewer control is needed which temporarily substituted for the Verification control to avoid user interaction on the document during Batch Edit.




For further information see the code sample BatchVerification.

If you have Thin Client Server installed, you can also use Thin Client Verification to validate a batch. Note that you have to enable thin clients and especially the Enable Thin Client Validation setting in the project settings in order to process a batch with Thin Client Verification. For further information, see *Kofax Transformation Toolkit - Thin Client Server Developer's Guide* .

### Gather statistics from a batch

When gathering statistics from a batch the same concept is applied as described in the section Gather Document Statistics. The only difference is that the documents used for statistical analysis are retrieved from the loaded `XFolder` object. The folder structure itself is not considered for statistical analysis.

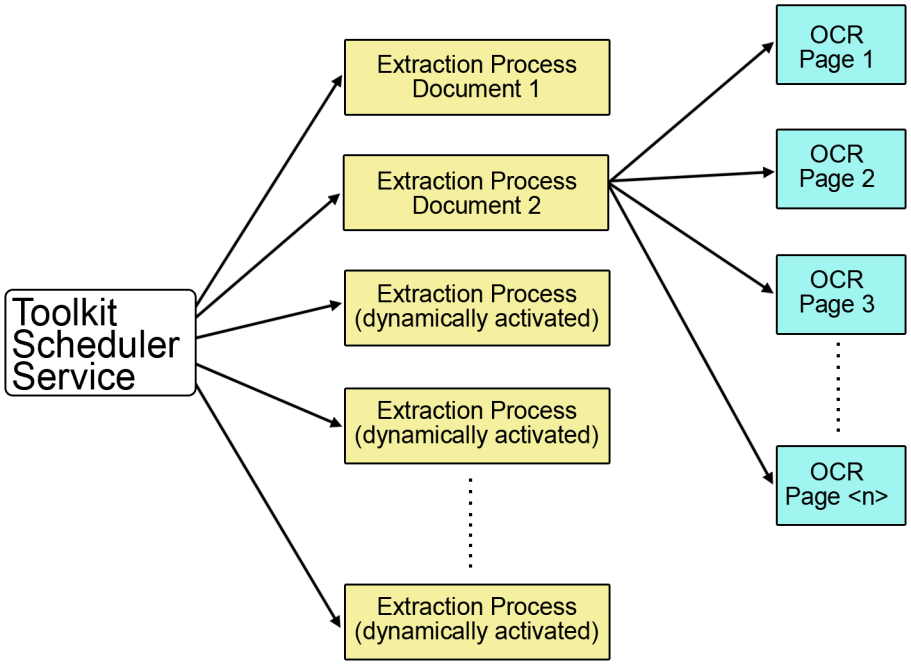
For further information see the code sample StatisticsServer.

## Use the Server Scheduler Service for parallelization

This chapter provides conceptual information about the Server Scheduler Service. This service allows the parallelization of extraction processes that means all kind of processing steps defined in the project (fpr).

The Server Scheduler Service allows utilizing a multi-core system during server processing. Depending on the document structure, pages or documents are processed in parallel to utilize the hardware as efficiently as possible.

The following graphic shows various processing steps the Server Scheduler Service exectues in a most efficient way using parallelization, especially to perform full text OCR that are the most time consuming processes.




For more details about how to the samples using the Server Scheduler Service work and the Server Scheduler Service API program listing, see Source Code Samples Using the Server Scheduler Service.

## Kofax Search and Matching Server

The Kofax Search and Matching Server is a server application that allows to set up and maintain fuzzy databases for large volumes of data (for example, customer or supplier databases). Various Kofax Transformation Toolkit applications can send search requests that are handled by the server in a most efficient way to perform non-exact (fuzzy) searches on those fuzzy databases. This means that search results are returned even if the search string contains spelling mistakes or does not match the exact value that is stored in the database.

In parallel to the search requests the server can perform scheduled database updates. For more information, refer to the *Kofax Search and Matching Server Getting Started Guide*.

The Kofax Transformation Toolkit installation contains an additional setup for the Kofax Search and Matching Server. It consists of a full installer that runs only on 64-bit operating systems and the Administration, a configuration tool for the Kofax Search and Matching Server that can run on 32-bit operating systems. A merge module is available that provides the interfaces to connect to the set up databases at a remote site. For more information, refer to the *Kofax Search and Matching Server Installation Guide*.

# Kofax Reporting

You can use Kofax Transformation Toolkit together with Kofax Reporting, a server application that allows gathering statistical data from various Kofax Transformation Toolkit applications. Statistics that are reported from various client sites to a central reporting site. The data is stored in a database that allows creating various type of reports to gain visibility about business processes as well as detailed information, for example, about the processing time and accuracy. For more information about the reporting platform, see Kofax Reporting documentation at the Kofax Web site.

The reporting platform is considered to gather statistical data on batch level. Therefore all batch processing source code samples, such as Batch Server and Batch Document Review, show how reporting can be implemented by referencing newly-added reporting dlls that encapsulate the functionality provided by Kofax Reporting.

In order to be able to report any statistical data the Kofax Reporting client and server must be installed and configured correctly. The communication between client and server is either done by a web service architecture (WSA) or an enterprise service bus (ESB), such as Sonic. For more details about the reported data, installation issues and other restrictions, see *Kofax Reporting for Kofax Transformation Toolkit Getting Started Guide* and for information about how to set up and configure the server and the communication software on client site, see *Kofax Reporting Administrator's Guide*.

When installing the reporting components for Kofax Transformation Toolkit you can configure the URL to the receiver at server site (WSA Receiver) during the Kofax Transformation Toolkit installation even if the receiver is not installed yet. Of course, the Kofax Reporting platform, including the communication software, must be fully installed and configured before any reported data can be written to the reporting database.

Especially, if you are installing the Kofax Transformation Toolkit software on several client sites using the silent install it is best practice to set the `REPORTINGURL` and `REPORTING` properties. Otherwise, you have to run the configuration tool to set the URL manually on every client computer.

# Kofax Transformation Toolkit - Thin Client Server

The Kofax Transformation Toolkit - Thin Client Server is a web application that allows processing batches for correction, validation and verification using Kofax Thin Client modules.

To process batches with a thin client module you need to install Kofax Transformation Toolkit - Thin Client Server containing the FileAccessBackend sample and configure the Microsoft Internet Information Service (IIS). For further details, see *Kofax Transformation Toolkit - Thin Client Server Developer's Guide* . The FileAccessBackend sample shows by using the file system how to process batches by the Thin Client modules.

You can use Thin Client Server together with Kofax Reporting, a server application that allows gathering statistical data from various Kofax Transformation Toolkit applications. Thin Client Server gathers statistical data and sends it to the reporting platform in case reporting is installed and

configured for Kofax Transformation Toolkit. The FileAccessBackend sample shows how to set batch details for reporting, but besides that no further implementation or configuration is needed as this is implemented as part of the central server process. If reporting is not configured no data is gathered and reported.

Chapter 5

# Source code samples

Kofax Transformation Toolkit comes with several sample projects, compatible with Microsoft Visual Studio 2010, which can be used to guide you in creating your own solutions. The following samples are available as examples of integration for the Kofax Transformation Toolkit component suite and are located in the following folder:

```
<Program Files>\Kofax\Transformation Toolkit\Resources\Examples
```

All samples that process batches, such as Batch Server and Batch Validation sample, send statistical data to the Kofax Reporting platform in case reporting is configured at server as well as at client site.

- Document Server sample - shows how to perform full text OCR on documents and extract data for the configured document types.
- Document Validation sample - shows how to use the Validation and `BatchView` controls to correct the extraction results for documents that are processed by the Document Server.
- Document Verification sample - shows for a set of documents how to use the functionality provided by the Verification control to verify fields.
- Batch Server sample - shows how to process batches by performing full text OCR, classification and extraction for the document types and processing methods defined in the project.
- Batch Correction sample - shows how to use the Correction control functionality to correct unconfident or missing OCR results for fields that are extracted using the Advanced Zone Locator for documents in a batch.
- Batch Document Review sample - shows how to use the `Document Review` control to correct document separation and classification results for a batch. Note that you need to process a batch using the Batch Processor .
- Batch Validation sample - shows how to use the Validation, `Document Viewer` and `BatchView` controls to correct the extraction results for documents of a batch that is processed by the Batch Server.
- Batch Verification sample - can be used to verify fields for a batch.
- Knowledge Base Learning Server sample - shows how documents that are marked for generic or specific online learning are processed so that they can be returned to update and improve the corresponding project and are added to the dynamic knowledge base. This sample supports document as well as batch processing.
- Statistics Server sample - shows how to gather statistics that can be viewed using the Statistics Viewer.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

## Limitations of the source code samples

The samples are intended to give an introduction in how to use the Kofax Transformation Toolkit components. They are not complete as various aspects are not considered.

**Batch state not set correctly**

The samples do not show the correct handling of the batch state, which can be "ready", "suspended", "completed" and "error". For the workflow you may want to ensure that you can proceed to a next module only in case all documents are completed for the current module. Therefore you have to set the state correctly, else documents are set to completed although there are, for example, problems or invalid fields remaining. Currently, if the application is closed the batch state is set "ready" and that means the document status for all documents of that batch are set to "completed".

**Batch state for last module in the workflow**

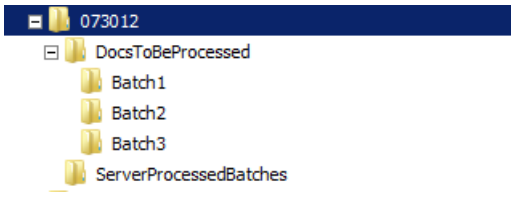
Currently, all samples set the batch status to "ready". In a workflow the last module in the workflow should set the batch state to "completed" to indicate that the processing for a batch is completed.

## Batch Server sample

This demo application processes batches of documents. That means a batch is a directory that contains a list of image (TIFF, JPEG, PNG) files. To process those batches you need to create an input folder that contains the batches that are processed. The BatchServer sample processes all batches with all its documents of the defined input folder and stores the results accordingly in a subfolder of the defined destination folder. After processing the batch of the destination subfolder contains a folder.xfd file that holds the structure information of the batch, the original image files and for each image a corresponding XDocument (.xdc) and a data file (text file with extraction results).

ⓘ For classification online learning, the loading of classifiers is suppressed when the project is loaded. Load the classifiers separately using CscClassifierLoadLocation.FromOLPathWithFallback. See an example in the sample code in the ValidateSettings method.

The following graphics shows an example of an input folder ("DocsToBeProcessed") that contains three batches to be processed. The Batch Server sample processes in a first step all documents contained in folder "Batch1" then all documents from "Batch2" and finally those located in "Batch3." After processing the output folder ("ServerProcessedBatches") contains three subfolders, Batch1-3 that are moved from the input folder containing additional files as described above. Those batches can now be processed by other applications such as Batch Document Review or Batch Validation.




The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

### Main step order demonstrated in the sample BatchServer

| Description | Location in Example Code |
|---|---|
| License the required components for the BatchServer.<br>On every computer a license must be activated using the Kofax License Utility. | See modMain.InitLicensing |
| Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain_Load and refer to `m_oReportingSession` object |
| Check if the classifiers have been updated and need to reload. | See frmMain.ReloadClassifiersIfNeeded |
| Initialize an XFolder object and fill it with documents from a directory path (depending on settings recursively or not.) | See frmMain.InitXFolder |
| Start the reporting session for the batch and its documents. | See frmMain.ProcessBatch and refer to `m_oReportingSession` object |
| Report user-defined actions during the `Batch_Open` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchOpen` batch processing is terminated and the batch is closed. | See frmMain.ProcessBatch and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchOpen` and `m_oReportingSession.EndBulkUpdate`<br>For a correct handling each `StartBulkUpdate` must be followed by `EndBulkUpdate`. Therefore `EndBulkUpdate` must be called for any additional error handling steps. |
| Process all documents of that XFolder object. | See frmMain.InternalProcessAllDocuments |

| Description | Location in Example Code |
|---|---|
| For each document of the batch the classification and extraction result and additional information, such as the time needed to classify the document, is reported.<br>During this processing step the XDocument is loaded in memory. | See frmMain.ProcessFile and refer to `m_oReportingSession` object |
| If Foldering is enabled: Report user-defined actions during the `DoFoldering` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventPerformAutoFoldering` batch processing is continued. | See frmMain.ProcessBatch and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventPerformAutoFoldering` and `m_oReportingSession.EndBulkUpdate` |
| Report user-defined actions during the `Batch_Close` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchClose` batch processing is terminated and the batch is closed. | See frmMain.ProcessBatch and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchClose` and `m_oReportingSession.EndBulkUpdate` |
| Close reporting session for the batch. | See frmMain.ProcessBatch and refer to `m_oReportingSession` object |
| Terminate the reporting session for this application. | See frmMain_Closing and refer to `m_oReportingSession` object |

For further details see Process A Batch.

# Document Server Sample

This demo application processes any images located in a configurable folder. The retrieved data is saved to XDocument files (.xdc) and stored in a destination folder. This folder also contains the source file (copy of original source file) and a data file (text file with extraction results).

The batch based samples are the only samples that support JPEG and PNG images. This sample only supports *.tif, *.txt, and *.pdf file formats.

**Main steps demonstrated in the DocumentServer sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | License the required components | See frmMain - InitLicensing function |
| 2 | Load a project file from disk into a project object | See frmMain – ValidateSettings |
| 3 | Create an XDocument | See frmMain – ProcessFile |
| 4 | Use XDoc.Import to set the source file in the XDocument | See frmMain – ProcessFile |
| 5 | Classify the XDocument using the Project.ClassifyXDoc Function | See frmMain – ProcessFile |
| 6 | Perform OCR on the XDocument using MpsPageRecognizing.Recognize function | See frmMain – PerformOCR |
| 7 | Extract fields using Class.Extract method | See frmMain – ExtractDoc |
| 8 | Save the XDocument | See frmMain – ProcessFile |

For further details see Classify And Extract A Single Document.

## Batch Document Review sample

This application demonstrates the integration of the Document Review control in Visual Basic .NET. A project containing document review rules can be loaded and used to validate an XFolder that contains documents which are previously classified. The application is used to validate document separation and classification results prior to extraction. In addition, you can also define batch and document rules that are raised as problems and can be fixed using the Document Review control.

The Document Review sample is considered to be used together with the Batch Processor sample. The Batch Processor can process document separation and can perform server processing in two separate steps. A first server step performs classification and document separation. Any problems can be reviewed and fixed by the Batch Document Review sample application. A second server step retrieves the data from the documents by performing extraction and applying validation rules and foldering.

To run this sample you need to start the test application and open the project that is used for the first server step. When you open a processed batch (folder.xfd) the Document Review control highlights all classification and document separation problems as well as failed document review rules. After all problems are solved you can close the batch and the project so that the second server step can be performed.

The sample does not show the correct handling of the batch state, which can be ready, suspended, completed or error. The batch state "ready" means that the batch does not have any problems. All documents are confidently classified and no batch nor document rules are failed; else the state "suspended" needs to be applied. If any error occurs the state error needs to be applied.

The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

**Main step order demonstrated in the Batch Document Review sample**

| Description | Location in Example Code |
|---|---|
| License the required components for the Batch Document Review.<br>On every computer a license must be activated using the Kofax License Utility. | See modMain.InitObjects |
| Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain.New and refer to `m_oReportingSession` object |
| Load a project into the Document Review control from an .fpr file | See mnOpenProject_Click |
| Load an XFolder from an .xfd file into the Document Review control | See mnOpenXFolder_Click |
| Start the reporting session for the batch and its documents | See mnOpenXFolder_Click refer to `m_oReportingSession` object |
| Report Document Review specific information for batch and document problems and rules as well as document separation results prior to the review of the batch. | See mnOpenXFolder_Click and `m_oReportingSession.StartDocumentReview` |

| Description | Location in Example Code |
|---|---|
| Report user-defined actions during the `Batch_Open` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchOpen` batch processing is terminated and the batch is closed. | See mnOpenXFolder_Click and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchOpen` and `m_oReportingSession.EndBulkUpdate`<br>For a correct handling each `StartBulkUpdate` must be followed by `EndBulkUpdate`. Therefore `EndBulkUpdate` must be called for any additional error handling steps. |
| Start the review for the loaded batch to fix any problems that occurred for document separation or classification and other batch or document problems as described below. | See mnOpenXFolder_Click (`DocumentReview1.ReviewBatch`) |
| Copy selected item (Page or Document) using Copy Menu Item | See mnCopy_Click |
| Paste item in clipboard (Page or Document) using Paste Menu Item | See mnPaste_Click |
| Zoom in on the image displayed in the Document Viewer control using the Zoom In menu item | See mnZoomIn |
| Zoom out of the image displayed in the Document Viewer control using the Zoom Out menu item | See mnZoomOut |
| Show the best fit for a page in the Document Viewer control using the Best Fit menu item | See mnBestFit |
| Navigate to first document using the First Document menu item | See mnFirst_Click |
| Navigate to the previous document using the Previous Document menu item | See mnPrevious_Click |
| Navigate to the next document using the Next Document menu item | See mnNext_Click |
| Navigate to the last document using the Last Document menu item | See mnLast_Click |
| Navigate to the previous problem using the Previous Problem menu item | See mnPreviousProblem_Click |
| Navigate to the next problem using the Next Problem menu item | See mnNextProblem_Click |
| Delete a document using Delete Document menu item | See mnDeleteDocument_Click |
| Merge a document to the previous using the Merge to Previous menu item | See mnMergeToPrevious_Click |

| Description | Location in Example Code |
|---|---|
| Merge multiple selected documents using Merge menu item | See mnMerge_Click |
| Split a document using Split before Current Page menu item | See mnSplitbeforeCurrentPage_ Click |
| Move selected document to the top of the batch using Move to Top menu item | See mnMoveToBegin_Click |
| Move selected document up one position using Move Up menu item | See mnMoveUp_Click |
| Move selected document down one position using Move Down menu item | See frmMain.mnMoveDown_Click |
| Move selected document to the end of the batch using Move to Bottom menu item | See mnMoveToEnd_Click |
| Navigate through the batch by reacting to the DocumentSelectionChanged event of the Document Review control | See DocumentReview1_ DocumentSelectionChanged |
| Delete a page from a document using Delete Page menu item | See mnDeletePage_Click |
| Rotate a document anti-clockwise using the Rotate Left menu item | See mnRotateLeft_Click |
| Rotate a document clockwise using the Rotate Right menu item | See mnRotateRight_Click |
| Move selected page to the first position in containing document using Move First menu item | See mnuMovePageFirst_Click |
| Move selected page backward one position in containing document using Move Left menu item | See mnuMovePageLeft_Click |
| Move selected page forward one position in containing document using Move Right menu item | See mnuMovePageRight_Click |
| Move selected page to the last position in containing document using Move Last menu item | See mnuMovePageLast_Click |
| Navigate through the document pages by reacting to the PageSelectionChanged event of the Document Review control | See DocumentReview1_ PageSelectionChanged |
| Generate Custom Menus as defined in Project Builder | See DocumentReview1_ CustomMenuDefined |
| Fire custom defined scripts when user clicks on custom menu | See SubMenu_Click |
| After all problems are resolved close XFolder and remove it from the Document Review control | See mnCloseXFolder |

| Description | Location in Example Code |
|---|---|
| Report user-defined actions during the `Batch_Close` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchClose` batch processing is terminated and the batch is closed. | See frmMain.ProcessBatch and `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchClose` and `m_oReportingSession.EndBulkUpdate` |
| Report evaluated document separation results after the review process for the batch is completed | See mnOpenXFolder_Click and `m_oReportingSession` |
| Close reporting session for the batch. | See ProcessBatch, refer to `m_oReportingSession` object |
| Close the project using Close Project menu item | See mnCloseProject |
| Terminate the reporting session for this application. | See frmMain_FormClosing refer to `m_oReportingSession` object |

For further details see Review A Batch.

## Batch Correction sample

In the demo application for the Correction control in Visual Basic .NET, a project and an XFolder can be loaded, and any fields extracted using an Advanced Zone Locator or OCR Voting Evaluator can be corrected. The application shows the usage of the Correction control to process fields in two modes: grouped by field name across the whole batch, or grouped by individual document (then by field name).

The sample does not show the correct handling of the batch state, which can be ready, suspended, completed or error. The batch state "ready" means that all fields selected for Correction (extracted with the Advanced Zone Locator) for all documents in the batch have confident OCR results or are corrected by the user for all documents in the batch; else the state "suspended" needs to be applied. If any error occurs the state error needs to be applied.

The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

**Main steps demonstrated in the BatchCorrection sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | License the required components for the BatchCorrection | See frmMain.InitObjects |
| 2 | Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain_Load and refer to `m_oReportingSession` object |
| 3 | Load a project from an .fpr file and an XFolder from an .xfd file and initialize the Correction control | See frmMain.mnOpenProject_Click and mnOpenFolder_Click |
| 4 | Report that the batch is opened | See mnOpenFolder_Click (`m_oReportingSession.OpenBatch`)<br>No bulk update for `BatchOpen` event needed as the event is not fired. |
| 5 | Save the changes to the field(s) | See MenuSaveCurrent_Click, mnSaveFolder_Click, and CtlCorrection1_BatchElaborated |
| 6 | Perform AutoDupe | See mnInsertLastValue_Click |
| 7 | Accept OCR value of the field | See AcceptOCR_Click |
| 8 | Navigate through the batch | See mnPreviousField_Click and mnNextField_Click |
| 9 | Full Image mode and zoom manipulation | See mnFullImage_Click, mnZoomIn_Click, mnZoomOut_Click and mnBestFit_Click |
| 10 | Report corrected fields | See OnSaveFolder and refer to `m_oReportingSession` object |
| 11 | Close reporting session for the batch. | See mnCloseFolder_Click, frmMain_FormClosing, or CtlCorrection1_BatchElaborated and refer to `m_oReportingSession` object<br>No bulk update for `BatchClose` event needed as the event is not fired. |
| 12 | Terminate the reporting session for this application. | See frmMain_FormClosed, refer to `m_oReportingSession` object |

For further details see Correct A Batch.

# Batch Validation sample

This application demonstrates the integration of the batch view and validation control in Visual Basic .NET. In this demo, a project and an XFolder can be loaded and the documents can be

validated. The application shows the usage of the validation control and of the batch view control and how to combine them. The online learning feature is also available.

Multi-step validation scenarios are also supported in this sample application. If the project being loaded has more than one validation step defined, a window is displayed to select which step to perform. Depending on the selected step, the control displays a different validation form, depending on the definition in the project.

The sample does not show the correct handling of the batch state, which can be ready, suspended, completed or error. The batch state "ready" means that all documents in the batch are successfully validated. All documents are classified and all fields are valid; else the state "suspended" needs to be applied. If any error occurs the state error needs to be applied.

The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

**Main step order demonstrated in the BatchValidation sample**

| Description | Location in Example Code |
|---|---|
| License the required components for the BatchValidation | See frmMain.InitObjects |
| Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain_Load refer to `m_oReportingSession` object |
| Load a project (.fpr) and initialize the Validation control and the BatchView control for it | See mnOpenProject_Click |
| Load an XFolder (.xfd) file and initialize the BatchView control for it | See mnOpenFolder_Click |
| Report user-defined actions during the `Batch_Open` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchOpen` batch processing is terminated and the batch is closed. | See mnOpenFolder_Click and `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchOpen` and `m_oReportingSession.EndBulkUpdate`<br>For a correct handling each `StartBulkUpdate` must be followed by `EndBulkUpdate`. Therefore `EndBulkUpdate` must be called for any additional error handling steps. |
| Connect the BatchView control with the Validation control and initialize fields for reporting | See BatchView1_LoadDocument and refer to `m_oReportingSession` object |

| Description | Location in Example Code |
|---|---|
| Save the changes to the XDocument object in events of BatchView control and updates data for reporting | See BatchView1_SaveChangesTo XDocument, BatchView1_SaveXDocument and refer to `m_oReportingSession` object |
| Navigate through batch by reacting on `DocumentValid` event of the Validation control | See Validation1_DocumentValid |
| Use the batch editing functionality of the BatchView control, switch between the Validation control and the DocumentViewer control. | See implementation of frmMain.BatchView1_... events |
| Execute Foldering again from within the Validation application | See DoAutoFoldering |
| If Foldering is enabled: Report user-defined actions during the `DoFoldering` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventPerformAutoFoldering` batch processing is continued. | See DoAutoFoldering and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventPerformAutoFoldering` and `m_oReportingSession.EndBulkUpdate` |
| Report user-defined actions during the `Batch_Close` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchClose` batch processing is terminated and the batch is closed. | See mnCloseFolder_Click and refer to `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchClose` and `m_oReportingSession.EndBulkUpdate` |
| Close reporting session for the batch. | See rmMain_FormClosing and refer to `m_oReportingSession` object |
| Terminate the reporting session for this application. | See frmMain_FormClosed and refer to `m_oReportingSession` object |

For further details see Validate A Batch.

# Batch Verification sample

This application demonstrates the integration of the batch view and validation control in Visual Basic .NET. In this demo, a project and an XFolder can be loaded and the documents can be

validated. The application shows the usage of the validation control and of the batch view control and how to combine them.

The sample does not show the correct handling of the batch state, which can be ready, suspended, completed or error. The batch state is "ready" means that all documents in the batch are successfully verified, else the state "suspended" needs to be applied. If any error occurs the state error needs to be applied.

The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

### Main steps demonstrated in the BatchVerification sample

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | License the required components for the BatchVerification | See frmMain.InitObjects |
| 2 | Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain_Load and refer to `m_oReportingSession` object |
| 2 | Load a project (.fpr) file and initialize the Verification control and the BatchView control for it | See mnOpenProject_Click |
| 3 | Load an XFolder (.xfd) file and initialize the BatchView control for it | See mnOpenFolder_Click |
| 5 | Report user-defined actions during the `Batch_Open` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchOpen` batch processing is terminated and the batch is closed. | See mnOpenFolder_Click and `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchOpen` and `m_oReportingSession.EndBulkUpdate`<br>For a correct handling each `StartBulkUpdate` must be followed by `EndBulkUpdate`. Therefore `EndBulkUpdate` must be called for any additional error handling steps. |
| 4 | Connect the BatchView control with the Verification control | See BatchView1_LoadDocument |

| Step | Description | Location in Example Code |
|---|---|---|
| 5 | Save the changes to the XDocument object in events of BatchView control and updates data for reporting | See BatchView1_SaveChangesTo XDocument, BatchView1_SaveXDocument and refer to `m_oReportingSession` object |
| 6 | Navigate through batch by reacting on DocumentVerified event of the Verification control | See Verification1_DocumentVerified |
| 7 | Use the batch editing functionality of the BatchView control, switch between the Verification control and the DocumentViewer control. | See implementation of BatchView1_... events |
| 12 | Report user-defined actions during the `Batch_Close` event implemented via script. All changes are send at once to the reporting platform after the `EndBulkUpdate` statement is performed. The `ModifiedInfo` object returns the applied changes. This is needed to update data that is reported accordingly.<br>If an error occurs when performing `RaiseEventBatchClose` batch processing is terminated and the batch is closed. | See mnCloseFolder_Click and `m_oReportingSession.StartBulkUpdate`, `RaiseEventBatchClose` and `m_oReportingSession.EndBulkUpdate` |
| 12 | Terminate the reporting session for this application. | See frmMain_FormClosed and refer to `m_oReportingSession` object |

For further details see Verify A Batch.

# Document Validation Sample

This application demonstrates the integration of the Validation control in Visual Basic .NET. In this demo, a project and an XDocument can be loaded and validated. The application shows the usage of the functionality of the Validation control and of the online learning feature.

The batch based samples are the only samples that support JPEG and PNG images. This sample only supports *.tif, *.txt, and *.pdf file formats.

This control can also be used in a multi-step validation scenario. If the project being loaded has more than one validation step defined, you should display a window or some other means to select which step to perform. Depending on the selected step, the control displays a different validation form, depending on the definition in the project.

**Main steps demonstrated in the Document Validation sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | Activate licensing of the required components | See modMain.InitObjects |
| 2 | Set user settings on the Validation control | See frmMain_Load |
| 3 | Load a project into the Validation control from an .fpr file | See mnOpenProject_Click |
| 4 | Load an XDocument from an xdc file into the Validation control | See mnOpenDocument_Click |
| 5 | If the project has multiple validation steps defined, present a window or some other method to select which step to perform. | See mnOpenDocument_Click |
| 6 | After user interaction with Validation control save the XDocument | See mnCloseDocument_Click |
| 7 | Mark an XDocument for Generic Online Learning | See mnOLGeneric_Click |
| 8 | Mark an XDocument for Specific Online Learning | See mnOLSpecific_Click |

For further details see Validate A Document.

# Document Verification Sample

This application demonstrates the integration of the Verification control in Visual Basic .NET. In this demo, a project and an XDocument can be loaded and the document verified. The application shows the usage of the functionality of the Verification control.

The batch based samples are the only samples that support JPEG and PNG images. This sample only supports *.tif, *.txt, and *.pdf file formats.

**Main steps demonstrated in the Document Verification sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | Activate licensing of the required components | See modMain.InitObjects |
| 2 | Set user settings on the Verification control | See frmMain_Load |
| 3 | Load a project (.fpr) into the Verification control | See mnOpenProject_Click |
| 4 | Load an XDocument (.xdc) into the Verification control | See mnOpenDocument_Click |
| 5 | After user interaction with Verification control save the XDocument changes | See mnCloseDocument_Click |

For further details see Verify A Document.

# Knowledge Base Learning Server sample

This application demonstrates how to handle the documents marked for generic and specific online learning. If marked for online learning the documents are copied to the "online learning folder" that is configured in the project so that they can be imported into the project to improve the extraction results. In addition, documents that are marked for specific online learning are added to the dynamic knowledge base that is used for the next processed batch. The application can be configured to process either single documents or a batch.

**Steps demonstrated in the KnowledgeBaseLearningServer sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | License the required components for the KnowledgeBaseLearningServer | See modMain.InitLicensing |
| 2 | Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See frmMain_Load refer to `m_oReportingSession` object |
| 3 | Load a project file from disk into a project class | See frmMain.ValidateSettings (called from frmMain_Load) |
| 4 | Initialize the `OnlineLearningManager` and init a batch | See frmMain.ProcessDocuments |
| 5 | Process an XDocument for Online Learning | See frmMain.ProcessFile -> ProcessOnlineLearning |
| 6 | Deinitialize the batch in the `OnlineLearningManager` object | See frmMain.ProcessDocuments |
| 7 | Close reporting session for the batch. | See rmMain.InternalProcessFolder refer to `m_oReportingSession` object<br>No bulk update for `BatchClose` event needed as the event is not fired. |
| 8 | Terminate the reporting session for this application. | See frmMain_FormClosing refer to `m_oReportingSession` object |

For further details see Process document for online learning and Process a batch for online learning.

# Statistics Server sample

Demo application that retrieves statistical information gathered while processing the documents in the server and validation applications. You can use the Statistics Viewer application to show various reports for the processed documents and batches.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

ⓘ You can skip the gathering of statistical data for a field by clearing the field's property "Use for statistics" that is selected by default.

**Steps demonstrated in the sample StatisticsServer**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | License the required components | See frmMain.InitObjects |
| 2 | Compact the database regularly | See frmMain.OpenStatsDatabase |
| 3 | Opens a statistics database (if required, a database is created) and initializes a data set for the processed batch | See frmMain.BatchOpen |
| 4 | Add statistics for a document with grouping information | See frmMain.ProcessDocument |
| 5 | Write statistics for batch | See frmMain.BatchClose |
| 6 | Aggregate the detailed data to aggregated historical data sets | See frmMain.BatchClose |
| 7 | Close statistics database | See frmMain.BatchClose |

For further details see Gather document statistics and Gather statistics from a batch.

Chapter 6

# Source code samples using the Server Scheduler Service

There are currently two source code samples that show how to process documents or batches using the Server Scheduler Service that allows extraction process parallelization.

The source code sample are not complete and may contain limitations.

The Server Scheduler Service is an optional feature. To install the service select "Toolkit Scheduler Service" feature during installation in the custom setup window. To install the source code samples that use the Server Scheduler Service the "Examples Using Scheduler Service" feature needs to be selected from the Custom Setup window.

The source code samples are written with Microsoft Visual Studio 2010 and are located in the following folder:

```
<program files>\Kofax\Transformation Toolkit\Resources\Examples
```

- ServiceUsingScheduler - sample for document processing
- BatchProcessor - sample for batch processing

## Programming API for Server Scheduler Service

The following listing shows the Microsoft Windows Communication Foundation (WCF) interface of the Server Scheduler Service.

```
namespace Kofax.Server.Scheduler.Interface
{
 /// <summary>
 /// Server scheduler interface
 /// </summary>
 /// <param name="Info"/>
 [OperationContract]
 void InitializeProcess(BatchProcessingInfo Info);

 /// <summary>
 /// Request the batch processing process to process one batch.
 /// </summary>
 [OperationContract]
 void StartProcessNextBatch();

 /// <summary>
 /// Wait for the completion of the currently processed batch.
 /// </summary>
```

```
 /// <param name="WaitTimout">Defines a timeout how long the function should wait for
 the completion.</param>
 /// <returns>eBatchCompletionState - see enum documentation above</returns>
 [OperationContract]
 eBatchCompletionState WaitForBatchCompletion(TimeSpan WaitTimout);

 /// <summary>
 /// Tell the batch processing process to cancel/suspend the current batch.
 /// It is still necessary to wait for the completion of the current batch with
 WaitForBatchCompletion
 /// </summary>
 [OperationContract]
 void CancelActiveBatch();

 /// <summary>
 /// Can be used to terminate the batch processing process.
 /// No other function can be called after this function call.
 /// </summary>
 [OperationContract]
 void QuitProcess();

}
```

The `ConsumeSchedulerService` class allows using the scheduler service easily. It provides all methods necessary to send batches to the scheduler.

```
namespace Kofax.Server.Scheduler.Interface
{
    /// <summary>
    /// A helperclass to ease the use of the scheduler service.
    /// </summary>
 public class ConsumeSchedulerService
 {
        /// <summary>
        /// Returns the session ID of the running scheduler service.
        /// </summary>
  public int SessionID{}

        /// <summary>
        /// Opens tcp channel to scheduler service.
        /// </summary>
  public ConsumeSchedulerService(){}

  /// <summary>
  /// Creates a session for the given project.
  /// </summary>
  /// <param name="ProjectName">Project filename including path.</param>
        public void CreateSession(string ProjectName){}

        /// <summary>
        /// Closes the current session and resets the SessionID to -1.
        /// </summary>
  public void CloseSession(){}

        /// <summary>
        /// Cancels this session, cleans up the queue and removes all unprocessed tasks
 from this session ID.
        /// </summary>
  public void CancelSession(){}

        /// <summary>
        /// Initializes the scheduler service with the given settings object.
        /// </summary>
```

```
        /// <param name="InitSchedulerObject">Object containing the settings to be
 applied for processing.</param>
  public void Initialize(InitSchedulerObject InitSchedulerObject){}

        /// <summary>
        /// Deinitializes the scheduler service.
        /// </summary>
  public void Deinitialize(){}

        /// <summary>
        /// Set preliminary information about incoming batch, needed for correctly
 distributing the system resources.
        /// </summary>
        /// <param name="oBatchStructureInfo">Object containing meta information of the
 incoming batch.</param>
  public void SetBatchInfo(BatchStructureInfo oBatchStructureInfo){}

        /// <summary>
        /// Processes the given task.
        /// </summary>
        /// <param name="oSchedulerTask">Object containing all information about the
 job to do.</param>
  public void ProcessTask(SchedulerTask oSchedulerTask){}

        /// <summary>
        /// Waits for the completion of the task.
        /// </summary>
        /// <returns>The task that is completed.</returns>
  public SchedulerTask WaitForTaskCompletion(){}

        /// <summary>
        /// Defines the executable to be used for the document processing tasks.
        /// </summary>
        /// <param name="path"/>
  public void SetExtractionPath(string path){}

        /// <summary>
        /// Defines the executable to be used for the batch processing tasks.
        /// </summary>
        /// <param name="path"/>
  public void SetBatchProcessingPath(string path){}

        /// <summary>
        /// Returns the current session ID.
        /// </summary>
        /// <returns/>
  public int GetSessionID(){}

        /// <summary>
        /// Starts an asynchronious waiting thread.
        /// </summary>
        /// <param name="bReadSingleTask"/>
  public void StartAsyncWaitThread(bool bReadSingleTask){}

        /// <summary>
        /// Stops the asynchronous waiting thread.
        /// </summary>
  public void StopAsyncWaitThread(){}

        /// <summary>
        /// Waits asynchronously for the given task.
        /// </summary>
        /// <param name="Milliseconds">Milliseconds to wait for the task completion.</
param>
```

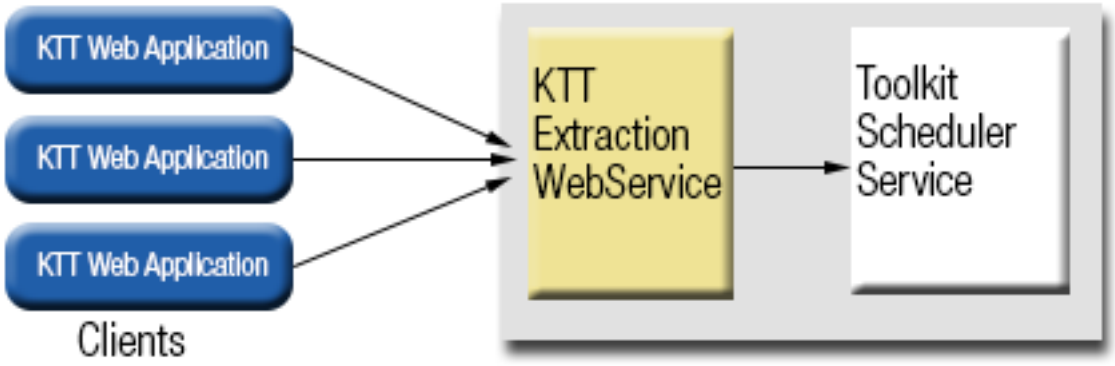
```
        /// <param name="oTask">Task to wait for.</param>
        /// <returns/>
  public bool AsyncWaitForTaskCompletion(int Milliseconds, ref SchedulerTask oTask){}
 }
}
```

Starting with 7.1.0 the Server Scheduler Service also supports configuration sets. A configuration set is assigned to the project and load specific settings related to the configuration are applied. Be aware that the introduction of this setting caused the `SchedulerTask` object interface to change. All `Init*` methods like `InitForProcessDocument` of the `SchedulerTask` have a new parameter for the name of the required configuration set.

## Document processing scheduler sample

The Kofax Transformation Toolkit Extraction WebService comes together with a sample application that demonstrates how to implement the service and how to use this web service interface.

The Kofax Transformation Toolkit Extraction WebService uses the Server Scheduler Service to classify and extract documents with a set of predefined projects. This functionality also includes the execution of full text OCR as configured in the used projects and the execution of the corresponding validation rules, which return the information if a document must be displayed to a user in the validation module.




The functionality of the Extraction WebService is restricted to document based methods (classification and extraction). Batch level script events or document separation functions are not supported in this version.

The Extraction WebService supports function calls over the network, which means the client application does not need to run on the same computer. While there is no technical limitation to prevent the usage of the Extraction WebService over the Internet, there is currently no security layer implemented that provides a basic authentication and authorization layer for a secure Internet access. Since the Kofax Transformation Toolkit Extraction WebService is provided as source code, it is possible to add any required security layer.

Kofax Transformation Toolkit setup provides a sample application for the Extraction WebService. To customize the application, use Microsoft Visual Studio 2010.

To run this source code sample ensure that the following steps are performed:

- The Extraction WebService must be installed. Select Visual Studio "installutil".
- The project root folder is configured in the registry. For more details about the registry path, see Configuring the Extraction WebService Project.
- If needed, change the port number in the registry in case the default port number (24333) is already used by another program or service. For more details about the registry path, see Configuring the Extraction WebService Port Number.

The batch based samples are the only samples that support JPEG and PNG images. This sample only supports *.tif, *.txt, and *.pdf file formats.

The following table shows the main steps to establish the Extraction WebService (Visual Studio project: KTT.ExtractionService):

**Main steps demonstrated in the Extraction WebService sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | Starting the server scheduler service if it is not running. | See clsSchedulerService.startServerScheduler |
| 2 | Initializing the server scheduler service | See clsSchedulerService.initScheduler |
| 3 | Task definition and calling the service scheduler service for processing a document with the specific task. | See clsSchedulerService.ExecuteProcessDocument |
| 4 | Implementation of the interface function for processing a document. Get a request message from a client and transform this message to a task that can be processed from the server scheduler service. | See ExtractionServiceImpl.ProcessDocument |
| 4a | Synchronous implementation of Step 4. | See ExtractionServiceImpl.ProcessDocument |
| 4b | Asynchronous implementation of Step 4. | See ExtractionServiceImpl.BeginProcessDocument ExtractionServiceImpl.EndProcessDocument |
| 5 | Retrieving the names of all available projects from the specified root folder. | See ExtractionServiceImpl.GetAvailableProjects |
| 6 | Retrieving the names of all available classes from the specified project. | See ExtractionServiceImpl.GetProjectClasses |

The sample application (Visual Studio project: KTT.WebService.TestApplication) processes documents located in a configurable directory. The retrieved data can be saved to either XDocument (.xdc) or

to XML files, which are placed in a destination directory along with the source file (copy of original source file). The used projects must be stored in a special project directory (only configurable with a registry key.)

**Main steps demonstrated in the Extraction WebService application sample**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | Starting the Extraction WebService if it is not running. | See MainForm.StartKTTExtractionService |
| 2 | Establishing a connection to the Extraction WebService on the specified computer. | See MainForm.Connect |
| 3 | Retrieving the list of all available projects from the Extraction WebService. | See MainForm.FillProjectsCombo |
| 4 | Retrieving the list of all available classes for the specified project from the Extraction WebService | See MainForm.FillClassesCombo |
| 5 | Sending the documents to the Extraction WebService for classification and extraction. This could be one document in the use case of sending a single document, or multiple documents in the use case of batch processing. The sample code shows the asynchronous method for document processing with the Extraction WebService. | See<br>• MainForm.ProcessSingleDocument<br>• MainForm.ProcessBatch<br>• MainForm.ProcessDocument<br>• MainForm.ProcessDocumentCallback |

## Configure the Extraction WebService project

The projects that are available for classification and extraction with the Kofax Transformation Toolkit Extraction WebService must be configured in a special project directory. This directory must be configured before starting theExtraction WebService. Each project used with the Extraction WebService must be stored in a separate subdirectory. The name of this subdirectory is used as the name of the project in the API.

To define the path of the folder that contains the project folders, you need use the following registry key:

1. From the **Start** menu, select **Run**.
2. Type "regedit" and press Enter.
   The registry editor is displayed.
3. Browse to the following registry key:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KTTExtractionService\Parameters
4. Create a string value that specifies the folder that contains all of the project folders.
   a. Right-click the right-hand pane, select New, and then select String Value.
   b. Name this string value "ProjectRootFolder".
   c. Double-click the string value.
      The Edit String window is displayed.
   d. In the **Value data** box, type in the path.

## Configure the Extraction WebService port number

For the Extraction WebService you can configure the port number and the project root folder in the registry. If the default port that is set to 24333 is already used by another program or service you can change the .

1. From the **Start** menu, select **Run**.
2. Type "regedit" and press Enter.
   The registry editor is displayed.
3. Browse to the following registry key:
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KTTExtractionService\Parameters
4. Create a string value that specifies the port number.
   a. Right-click the right-hand pane, select New, and then select String Value.
   b. Name this string value "Port".
   c. Double-click the string value.
      The Edit String window is displayed.
   d. In the **Value data** box, type the port number that you want to use for this service.
      The default port number is 24333.

## Programming API for Extraction WebService

The following listing shows the Microsoft Windows Communication Foundation (WCF) interface of the Extraction WebService.

```
namespace KTT.WebService.Interface
{
    [ServiceContract]
    public interface IKTTExtractionService
    {
        /// <summary>
        /// Function used to get projects names list from the specified location
        /// </summary>
        [OperationContract]
        List<String> GetAvailableProjects();

        /// <summary>
        ///  Function used to get classes names list of the specified project
        /// </summary>
        [OperationContract]
        List<String> GetProjectClasses(string projName);

        /// <summary>
        ///  Function used to process document and get streamed document as zip file
 including the source file and the xdoc
        /// </summary>
        [OperationContract]
            [FaultContract(typeof(KTTServiceException))]
        KTTExtractionResponseMessage ProcessDocument(KTTExtractionMessage message);

            /// <summary>
            /// Asynchronous processing of ProcessDocument
            /// </summary>
```

```
            [OperationContract(AsyncPattern = true)]
            IAsyncResult BeginProcessDocument(KTTExtractionMessage message,
 AsyncCallback callback, object asyncState);

            //Note: There is no OperationContractAttribute for the end method.
            [FaultContract(typeof(KTTServiceException))]
            KTTExtractionResponseMessage EndProcessDocument(IAsyncResult result);
      }
}
```

The `KTTExtractionMessage` defines the WCF message contract that is used as parameter for the `ProcessDocument` function. The input document is passed as a zip stream using the `SourceStream` parameter. The sample application contains the necessary source code to create a zip stream.

```
namespace KTT.WebService.Interface
{
    [MessageContract]
    public class KTTExtractionMessage
    {
            /// <summary>
            /// a unique identifier for this message
            /// </summary>
            [MessageHeader]
            public string Id { get; set; }

            /// <summary>
        /// a string that references the project name
        /// </summary>
        [MessageHeader]
        public string ProjectName { get; set; }

        /// <summary>
        /// a string that specifies the extraction class in case that only extraction
 should be performed
        /// </summary>
        [MessageHeader]
        public string ExtractionClass { get; set; }

        /// <summary>
        /// If this flag is true, the document is classified
        /// If the flag is false, the ExtractionClass property should contain the class
 name for extraction
        /// </summary>
        [MessageHeader]
        public bool DoClassification { get; set; }

        /// <summary>
        /// If this flag is true, the document is extracted
        /// </summary>
        [MessageHeader]
        public bool DoExtraction { get; set; }

        /// <summary>
        /// If this flag is true, the document is validated
        /// </summary>
        [MessageHeader]
        public bool DoValidation { get; set; }

        /// <summary>
        /// the streamed document as zip file including the source file and the xdoc.
        /// </summary>
        [MessageBodyMember]
```

```
        public System.IO.Stream SourceStream;

    }
}
```

The `KTTExtractionResponseMessage` defines the WCF message contract that is used as return value from the `ProcessDocument` function. The XDocument containing all the results is also passed as a zip stream using the `ResponseStream` parameter.

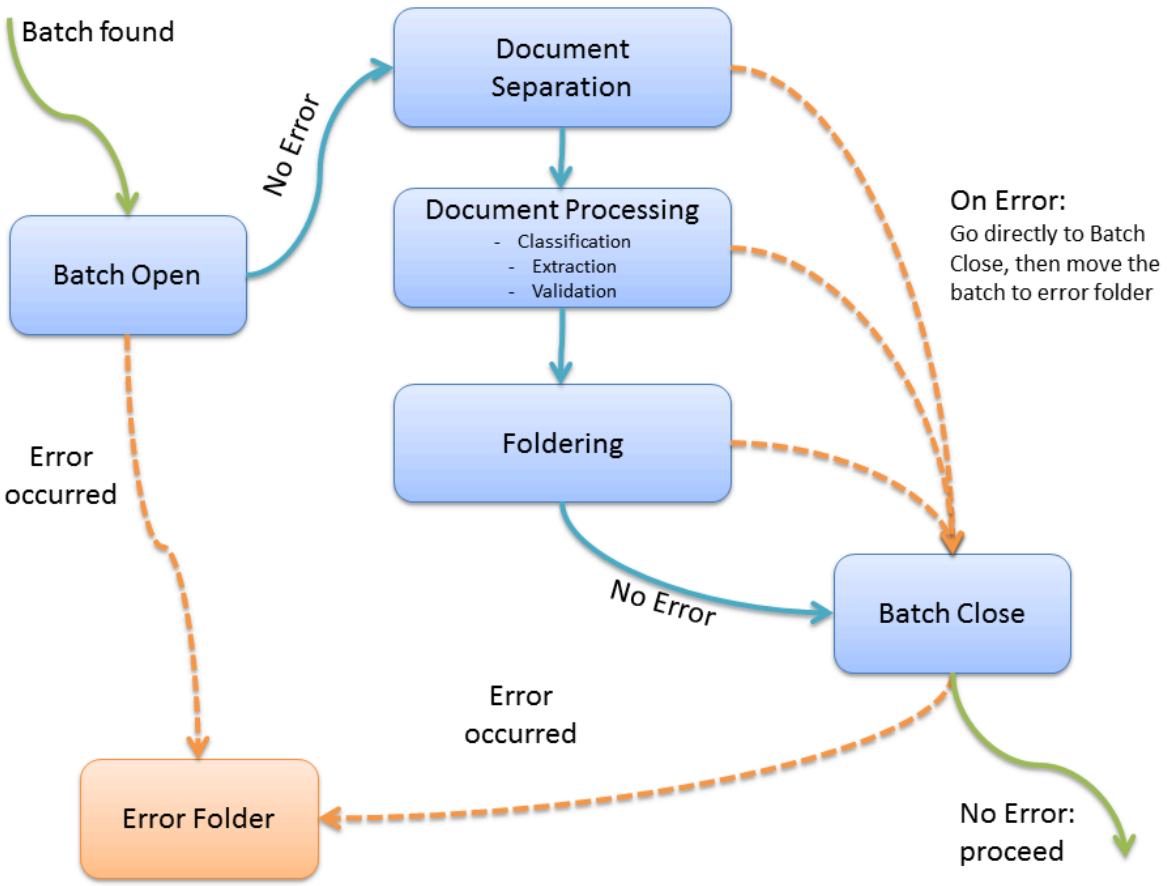
```
namespace KTT.WebService.Interface
{
    [MessageContract]
    public class KTTExtractionResponseMessage
    {
            /// <summary>
            /// a unique identifier for this message (copied from request message)
            /// </summary>
            [MessageHeader]
            public string Id { get; set; }

            /// <summary>
        /// the streamed document as zip file including the source file and the xdoc.
        /// </summary>
        [MessageBodyMember]
        public System.IO.Stream ResponseStream;
    }
}
```

# Batch processing with Server Scheduler Service (batch processor) sample

This sample demonstrates how to implement batch processing with the Server Scheduler Service.

The Batch Processor prepares the documents of various batches for the Server Scheduler Service. The scheduler then performs the document processing tasks that are send by the Batch Processor. The document processing tasks cover document separation, classification, extraction, executing validation rules and foldering.




The Batch Processor processes batches located in an input directory. The processed batches are then stored in a configured output directory. The Kofax Transformation project that defines the document separation, classification, extraction and validation steps that are done during the document processing must be stored in a folder. These settings can be configured in the Settings window of the Batch Processor application, which are then stored in the registry for the following key. These settings are also used when the Batch Processor is started from the command line. These settings can be found in the registry under `HKEY_CURRENT_USER`.

```
Software\Kofax\Toolkit\KTTBatchProcessor
```

These are the string values and their descriptions.

`ServerInstance1InputDirectory`

The location of the batches that are then processed by the Batch Processor for instance 1.

`ServerInstance1OutputDirectory`

The location of the processed batches for instance 1.

`ServerInstance2InputDirectory`

The location of the batches that are then processed by the Batch Processor for instance 2.

`ServerInstance2OutputDirectory`

The location of the processed batches for instance 2.

`ErrorDirectory`

The location to store batches that were not processed correctly.

`UseTwoInstances`

If the setting is TRUE processing is performed in two steps. Step 1 performs document separation and classification. In the second step, extraction, validation rules and foldering are performed. However, various processing steps are optional, such as document separation and foldering. With the help of the Project Builder application you can configure all needed processing steps and store them as a project file.

`ProjectFullFilename`

The location of the project file. The project file contains all configuration for processing the documents of a batch, for example, for the document separation, classification, extraction and foldering. These determine how the documents are processed.

`ProcessImages`

If set to TRUE all supported image files in the batch directory are processed.

`ProcessPdfs`

Set this value to TRUE when your batch directory contains PDF files that you want processed.

`ConfigSetName`

With Kofax Transformation Toolkit 7.1.0 the `BatchProcessor` also supports configuration sets. These sets configure the project and on load, specific settings related to the configuration are applied.

If adding the configuration name, the project will be loaded with the configuration set if it is available. With the introduction of this setting the interface of the Server Scheduler has changed.

Refer to the Project Builder help on configuration sets.

All `Init*` methods like `InitForProcessDocument` of the `SchedulerTask` has got an additional parameter for the name of the required configuration set. If you do not want to use configuration sets, just enter an empty string.

To process batches you can start the Batch Processor manually by either by running the test application or from the command line. To run the test application you need to execute the `KTT.BatchProcessorGUI.exe` in the Bin folder. The `KTT.BatchProcessorGUI.exe` provides a user interface that allows configuring the necessary settings and starting the batch processing.

To call the `KTT.BatchProcessorCLI.exe` from a command line type `run <installdir>\Bin \KTT.BatchProcessorCLI.exe TestMode`. The `KTT.BatchProcessorCLI.exe` starts the Batch Processor without a user interface and retrieves the settings, such as the location of the project file, the input and output directory and the error directory, from the registry. Note that these settings must be configured in order for the `BatchProcessorCLI` to run.

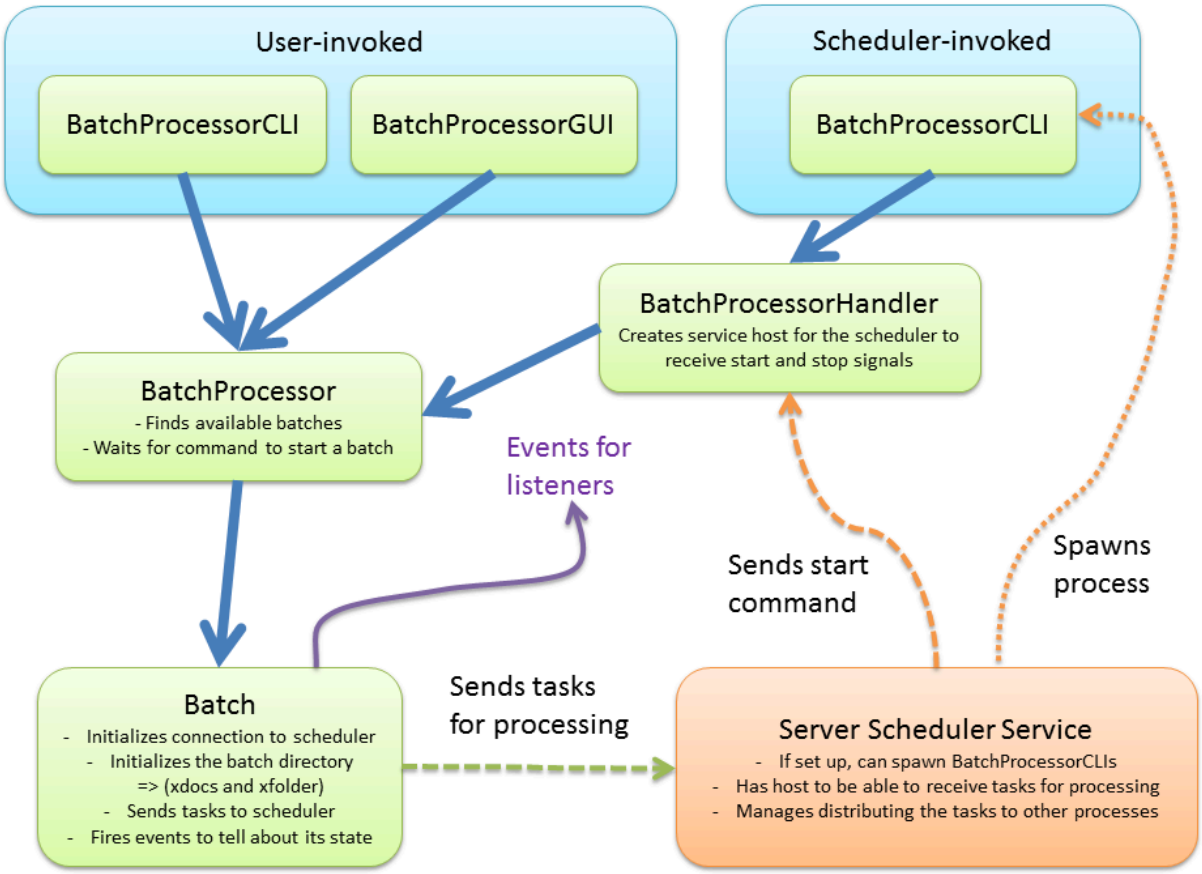
In addition to the user-invoked batch processing the Server Scheduler Service can be configured to start various `KTT.BatchProcessorCLIs` automatically when automatic processing is enabled for the scheduler. This is done by setting the value to "1" for the following registry key.

`HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Kofax\Transformation\4.0\Server\AutoStartBatchProcessing`

The demo application contains sample code that shows how to implement gathering statistical data for the Kofax Reporting platform.

The reporting references as well as the `amd` and `x86` subfolders must be located in the `bin` folder of any sample application that is supposed to use Kofax Reporting or in the folder where the application executable is located. For additional information about the Kofax Reporting platform and the required installation, see *Kofax Reporting Administrator's Guide*.

Below the following diagram that shows the different ways to invoke the processing at first the main steps for the user-invoked processing are described, and then the main steps for the scheduler-invoked processing.




The following table shows the main steps to demonstrate the user-invoked batch processing.

**Main step order demonstrated in the user-invoked processing sample**

| Description | Location in Example Code |
|---|---|
| Initialize the batch processor instance | See KTT.BatchProcessor.Init |
| Create and initialize reporting session; information such as the application and user name are sent to the reporting platform. | See KTT.BatchProcessor.BatchProcessor refer to `_ReportingSession` object |
| Start processing the next batch | See KTT.BatchProcessor.ProcessBatch |
| Collect configuration parameters for the selected batch. The batch object is instantiated | See KTT.BatchProcessor.StartBatchProcess |
| Start processing and reporting for the batch and its documents | See KTT.Batch.Run refer to `_ReportingSession` object |
| Load the project file and initializes the XFolder object and the corresponding XDocument files (*.xdc) | See KTT.Batch.Init |
| Connect to the Server Scheduler Service | See KTT.Batch.ConnectToScheduler |
| Send a `BatchOpen` task to the scheduler | See KTT.Batch.RequestBatchOpen |
| Send the task "document separation" to the scheduler, if configured in the project. Report results of document separation (number of correct splits) | See KTT.Batch.RequestDocumentSeparation (`_reportingSession.Separate` and `_reportingSession.FinalizeSeparation`) |
| Send one task to the scheduler for each document to perform classification, extraction and validation rules. The scheduler can parallelize these tasks | See KTT.Batch.RequestDocumentProcessing |
| Report results of document processing | See KTT.Batch.ProcessResults (`_reportingSession.Process`) |
| If foldering is configured in the project, send the task "foldering" to the scheduler and report results. | See KTT.Batch.RequestFoldering (`_reportingSession.ReportActionItems`) |
| Send a `BatchClose` task to the scheduler and report actions | See KTT.Batch.RequestBatchClose (`_reportingSession.ReportActionItems`) |
| If processing is successful the batch is moved to the output directory. In case of an error the batch is moved the error folder. | See KTT.Batch.MoveToFolder |
| Close reporting session for the batch. | See frmMain.ProcessBatch refer to `m_oReportingSession` object |
| Terminate the reporting session for this application. | See frmMain_Closing refer to `m_oReportingSession` object |
| Disconnect from the Server Scheduler Service. | See KTT.Batch.Disconnect |

You can also run the Server Scheduler Service in an automatic processing mode. Then the scheduler can use its load balancer in order to process batches in a most effective way. In general, most processing time is used for OCR. In case of small batches where each batch consists of one document, OCR cannot be parallelized, because one Batch Processor handles one batch

only and parallelization happens for the documents in one batch. In order to use all available system resources the scheduler tries to process many batches in parallel and starts several KTT.BatchProcessorCLIs. For large batches one Batch Processor is sufficient to provide many images that need OCR at the same time for the scheduler.

For automatic processing the scheduler spawns a `KTT.BatchProcessorCLI.exe` that then creates a service host via Microsoft Windows Communication Foundation (WCF) in the `BatchProcessorHandler` class. The scheduler contacts the service host to request a new batch or to send batches continuously or to quit a running batch.

The following table shows the main steps for batch processing that is invoked by the Server Scheduler Service.

**Main steps for the scheduler-invoked batch processing**

| Step | Description | Location in Example Code |
|---|---|---|
| 1 | Set the registry key for automatic processing to "1" and restart the Server Scheduler Service to load the new settings. | |
| 2 | Opens a service host endpoint of type BatchProcessorHandler. | See KTT.BatchProcessorCLI.CallScheduler |
| 3 | The scheduler uses this endpoint to call `InitializeProcess` using the `IBatchProcessingProcess` interface. | See BatchProcessorHandler.InitializeProcess |
| 4 | Starts thread to run Batch Processor class in the background, to be open for input from the Server Scheduler Service. | See BatchProcessorHandler.StartProcessNextBatch |
| 5 | Initializes the Batch Processor and starts the next batch.<br>This corresponds to steps 1 and 2 of the user-invoked processing. The remaining steps are the same as above. | See BatchProcessorHandler.STAProcessBatchLoop |

The service endpoint offers the scheduler service all methods as defined in `Kofax.Server.Scheduler.Interface.IBatchProcessingProcess`. If used in this way the KTT.BatchProcessorCLI is not available for user input, but acts as a fully automated process.

## Batch processor application user interface

This Batch Processor application (KTTBatchProcessorGUI.exe) shows how to process one or more batches sequentially (no batch parallelization) using the Server Scheduler Service via a user interface. This section provides details on the Batch Processor application user interface.

The Batch Processor application has a main menu, statistics and progress indicators, as well as a current image pane and a status bar. While processing a batch, progress and certain statistics are displayed.

- Statistics and progress indicator panes
- Status bar

The status bar at the bottom of the screen provides status messages and other information.

- The Settings Window

The main menu has the following menu items:

- Batch
- Settings
- BatchMode

The Batch menu has the following menu items:

**Process selected batch**

Starts processing the selected batch in the Current Batch Information list.

**Continuous batch processing**

Starts processing the first batch in the Current Batch Information list then processes all following batches in the list.

**Stop current batch**

Stops processing the batch currently in progress.

**Reload batches**

Refreshes the Current Batch Information list for the batches located in the configured input directories for instance 1 and, if selected, also for instance 2.

**Exit**

Closes the Batch Processor application.

The Settings menu allows displaying the Settings window in order to set the needed configuration options.

The BatchMode menu provides the currently selected setting for the following menu items:

**Both**

By default Both is selected and determines that the Batch Processor processes all batches configured for instance 1 and 2. That means that a first server step is performed for the batch located in the input folder for instance 1, which includes document separation and classification. For instance 2 batches the second server step is performed, which means extraction, validation rules and foldering when configured.

This setting is available only if you have selected Use two instances on the Settings window.

**Server1**

Select Server1 to perform only the first server step including document separation and classification. All batches located in the configured input directory for instance 1 are processed.

If you configure instance 1 only on the Settings window the options Both and Server2 are not available in the menu. Then the server performs the complete processing in one server step and by default Server1 is selected. When you define two instances the processing is divided into two separate steps and you can review the result of the separation and classification before performing the extraction.

**Server2**

Select Server2 to perform only the second server step, including extraction, validation rules and foldering. All batches located in the configured input directory for instance 2 are processed.

This setting is available only if you have selected Use two instances on the Settings window.

## Statistics and progress indicators

The following indicators provide information about statistics and the progress of a batch.

**Current Batch Information**

This includes a list of batches depending on the configured batch mode. All batches that are located in the directory for Instance 1 and, if defined, for both instances 1 and 2 are displayed.

**Current Batch Progress**

This area shows the progress of the current batch measured by the number of documents processed and the number of remaining documents. A dynamic bar and a CPU usage history graph show how much CPU is being utilized along with all running extraction processes.

**Current Document Statistics**

This area shows statistics for the current document. It shows the total number of successfully extracted fields relative to all fields that are already processed and the average extraction confidence. The average confidence can be used to judge overall extraction quality.

**Current Document Information**

This area shows information about the current document, including the name of the current class (displayed in blue) as well as all fields and their extraction results.

**Event Log**

The Warning count displays the number of warnings and the Error count displays the number of errors found during document processing progress. Click **Reset Counters** to set the counters back to zero. The Event Log area displays all kinds of error messages and warnings that occurred during batch and document processing. To see a full message in a message box, double-click an entry.

**Current Image**

The Current Image pane displays the actual processed document as soon as extraction results are retrieved.

During production, the image is displayed for less than a second.

## Settings window

The Settings window is displayed when selecting Settings from the Batch Processor user interface main menu. It allows setting the following options.

The settings are also stored in the registry (`Software\Kofax\Toolkit\KTTBatchProcessor`)

The **General** group has the following menu items:

**Project file (.fpr)**
Select the location of the project file by either typing the full filename or clicking the button and navigating to the directory where the project file is located. The project file contains all configuration for processing the documents of a batch, for example, for the document separation, classification, extraction and foldering. These determine how the documents are processed.

**Error directory**
Select the location to store batches that were not processed correctly.

The Instance 1 group has the following menu items:

**Input directory**
Select the location of the batches that are then processed by the Batch Processor for instance 1.

**Output directory**
Select the location of the processed batches for instance 1.

The Instance 2 group has the following menu items:

**Use two instances**
If the setting is selected, you can configure the input and output directories for instance 2. That means that batches located in both input directories can be processed by the sample application and processing is performed in two steps. Step 1 performs documentation separation and classification. In the second step extraction, validation rules and foldering are performed. However, various processing steps are optional, such as document separation and foldering, and are configured in the corresponding project file. By default this setting is cleared and no folders need to be determined. In that case the complete processing is done in one processing step.

**Input directory**
Select the location of the batches that are then processed by the Batch Processor for instance 2.

**Output directory**
Select the location of the processed batches for instance 2.

The File Types group has the following menu items:

**Image**

By default this setting is selected so that all supported image files of the batch directory are processed.

**PDF**

Select this setting when your batch directory contains PDF documents that you want to process.

Chapter 7

# Considerations for integration

For redistribution several merge modules are available that provide the Kofax Transformation Toolkit functionality. Those need to be installed together with your product to provide the full functionality. In addition, there are two full setups provided to install the Kofax Search and Matching Server and the Kofax Transformation Toolkit - Thin Client Server.

The Kofax Search and Matching Server installer is a full installer that provides a server application for the remote access of fuzzy indexes. This server application cannot be customized.

For the Kofax Transformation Toolkit - Thin Client Server the FileAccessBackendSample dll used by the Thin Client Server to provide the available batches that are then processed by the thin client modules can be replaced by your own back-end integration that integrates your workflow to the Thin Clients modules. In that case you can use the provided Thin Client Server installation, run it from the command line using a parameter so that the Kofax FileAccessBackendSample dll is not installed and use MSI transform technology for further customization. In addition to the base Thin Clients installation you need to install your own integration.

```
KTS.msi NOEXAMPLES=1
```

For further information about the FileBackend sample, the customization of the thin client modules or the help, refer to *Kofax Transformation Toolkit - Thin Client Server Developer's Guide*.

## Merge modules

The following merge modules are part of Kofax Transformation Toolkit ISO image and located in the Redistributables folder. These merge modules can be included as part of your setup in order to install the Kofax Transformation Toolkit functionality at a customer site. To provide the complete functionality some merge modules are mandatory whereas others are optional and needed only in case specific functionality is going to be provided. The following list gives a description and recommendations for their installation.

**KofaxBasicReferencesV24.msm**

Contains basic libraries used by all applications.

Contains all references installed to the Global Assembly Cache that are needed by Kofax Transformation Toolkit.

It installs to the following location:

GlobalAssemblyCache (GAC)

**KofaxBrainwareComponentsV3.0.1.msm**

Installs the components for Brainware-related algorithms.

**KofaxCheckRecognition1.0.Data.msm**

Contains the Kofax Transformation Toolkit CheckPlus, CheckPlusInternational, CheckUltra, and CheckUsability recognition engine components. If you want to use this recognition engine, both this and the `KofaxCheckRecognition1.0.Sdk.msm` merge module are required. It installs to the following location:

`<Program Files>\Common Files\Kofax\KofaxCheckRecognition1.0`

**KofaxCheckRecognition1.0.Sdk.msm**

Contains the Kofax Transformation Toolkit CheckPlus, CheckPlusInternational, CheckUltra, and CheckUsability recognition engine components. If you want to use the check plus recognition engine, both this and the `KofaxCheckRecognition1.0.Data.msm` merge module are required. It installs to the following location:

`<Program Files>\Common Files\Kofax\CheckRecognition1.0`

**KofaxComponentInteropsV24.msm**

Contains the Interops for Kofax Transformation Toolkit components. It installs to the following location:

GlobalAssemblyCache (GAC)

**KofaxComponentsV24.msm**

Contains the Kofax Transformation Toolkit components. It installs to the following location:

`<Program Files>\Common Files\Kofax\Components`

**KofaxDatabaseCommonV24.msm**

Contains the database dialog component and components for online learning and statistics. It installs to the following location:

GlobalAssemblyCache (GAC)

`<Program Files>\Common Files\Kofax\Components`

**KofaxDetectHpMp_V3.0.msm**

Contains the Kofax Transformation Toolkit integrated DetectHpMp components for mixed print text recognition. It installs to the following location:

`<Program Files>\Common Files\Kofax\DetectMpHp`

**KofaxFormXtra7.6.Data.msm**

Contains part of the Kofax Transformation Toolkit integrated FormXtra recognition engine components. If you want to use the FormXtra recognition engine, both this and the `KofaxFormXtra7.6.Sdk.msm` merge module are required. It installs to the following location:

`<Program Files>\Common Files\Kofax\FormXtra76`

**KofaxFormXtra7.6.Sdk.msm**

Contains part of the Kofax Transformation Toolkit integrated FormXtra recognition engine components. If you want to use the FormXtra recognition engine, both this and the `KofaxFormXtra7.6.Data.msm` merge module are required. It installs to the following location:

`<Program Files>\Common Files\Kofax\FormXtra76`

**KofaxImageFilter.msm**

Contains the Image Filter for Kofax Transformation Toolkit components. It installs to the following location:

`<Program Files>\Common Files\Kofax\Components\Img`

**KofaxInfragistics11.1Basic.msm**

Contains the Infragistics .NET 11.1 basic components for Global Assembly Cache. It installs to the following location:

GlobalAssemblyCache (GAC)

**KofaxNLPV4.2.msm**

Contains the components needed by Kofax Transformation Toolkit for Natural Language Processing. It installs to the following location:

`<Program Files>\Common Files\Kofax\NLP`

**KofaxOmniPage22.0.0.3.msm**

Contains the Kofax Transformation Toolkit integrated OmniPage 22.0.0.3 recognition engine components. It installs to the following location:

`<Program Files>\Common Files\Kofax\OmniPage22`

**KofaxProjectBuilderV24.msm**

Contains the Project Builder application and other tools, dictionaries, Knowledge Bases, and language packs (for tables). You must specify the installation path when creating the installer.

This merge module is optional and required only if you want to install Project Builder.

**KofaxReportingReferencesV2.0.msm**

Contains Kofax Reporting components. In order to use the reporting functionality this merge module is mandatory and must always be installed.

**KofaxServerV24.msm**

Contains the Server Scheduler Service application, which supports parallel processing of documents. It installs to the following location:

GlobalAssemblyCache (GAC)

`<Program Files>\Common Files\Kofax\Server`

For a custom installation, you need to configure this merge module by setting the following properties to the required values, otherwise the default values are applied.

- `SCHEDULERLOGGINGPATH` - This is the path used by the Server Scheduler Service to log information, warnings and errors. **Default**:

  `CommonAppDataFolder\Kofax\ServerScheduler`
- `OEMPRODUCTNAME` - This is the product name used in the display name and the description of the Server Scheduler Service. **Default**: Kofax Transformation Toolkit

**KofaxToolkitLicensingV24.msm**

Contains the licensing client dlls required to communicate with the license server.

**KofaxStatisticsViewerV24.msm**

Contains the Statistics Viewer application. This merge module is installed to the same folder as the Project Builder application. You must specify the installation path when creating the installer.

This merge module is optional and required only if you want to install Statistics Viewer.

⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

**KofaxToolkitBasicV24.msm**

Contains the version information for Kofax Transformation Toolkit.

This merge module is mandatory and is installed in the GAC.

**KofaxToolkitInteropsV24.msm**

Contains references for development and if available, xml documentation files.

This merge module is optional and used for development only.

**KofaxToolkitLicensingServerV24.msm**

Contains the licensing for the Kofax Transformation Toolkit components. This merge module installs the licensing server files and registers the License Service. It installs to the following location:

GlobalAssemblyCache (GAC)

```
<Program Files>\Kofax\Licensing\Server
```

This merge module is mandatory, but the License Server does not need to be installed on the client site.

**KofaxToolkitLicensingV24.msm**

Contains the License Utility and the libraries required to access the hardware or software key. It installs to the following location:

GlobalAssemblyCache (GAC)

```
<Program Files>\Common Files\Kofax\Licensing
```

This merge module is mandatory and must always be installed.

**KofaxToolkitReportingConfigurationV2.0.msm**

Contains Kofax Reporting components to configure the connection to the WSA Receiver that stores statistical data in a reporting database. This merge module needs to be executed during the installation in order to install and run the reporting configuration tool that sets the URL to the WSA Receiver.

ⓘ You can run the configuration tool at a later point at time to change the configured settings.

It installs the configuration tool to the following location:

```
<Program Files>\Kofax\Kofax Toolkit Reporting
```

This merge module is optional, but mandatory when used together with Kofax Reporting. It needs to be executed during the installation so that it is installed on each client site. In order to use the Kofax Reporting platform several components need to be installed on client as well as at the server site. For more details, see *Kofax Reporting Administrator's Guide*.

**KofaxToolkitRuntimeV24.msm**

Contains the runtime components for Kofax Transformation Toolkit, including the control libraries for Correction, Document Review, Validation, and Verification. It installs to the following location:

```
<Program Files>\Common Files\Kofax\Toolkit
```

**KofaxWinWrapV9.2.msm**

Contains the WinWrap Basic Scripting Engine, including all available languages for the scripting control. It installs to the following location:

`SYSTEM32`

**KSALicClientMerge.msm**

This is a prerequisite for the `KofaxToolkitLicensingServerV06.10.msm` merge module. This merge module is mandatory and must always be installed.

**KSALicServerFilesOnlyMerge.msm**

This is a prerequisite for the `KofaxToolkitLicensingServerV06.10.msm` merge module.

# Customization

For Kofax Transformation Toolkit an integrator-specific "Kofax.OEM.Product.dll" is provided. If you are using a full setup for installation at client site, the Kofax Transformation Toolkit "Kofax.OEM.Product.dll" can be replaced by this integrator-specific .dll. The following locations are relevant:

- `<Program Files>\Common Files\Kofax\Server`
- `<Program Files>\<Location of Project Builder>`
- bin folder of all of your customized applications

If you create your own setup, please ensure that this integrator-specific .dll is present at all of the above locations.

German resources (de\Kofax.OEM.Product.resources.dll) are also available for this .dll.

## Warning when volume licenses exceed a limit

To ensure that Server processing does not stop for a production system when the volume licenses for OCR, classification and extraction exceed their limit, the hardware or software key normally provides one month's worth of processing volume as an additional single-use volume license. This single-use volume is depleted when the other volume licenses have expired.

In order to inform the customer when the volume licenses have exceeded a limit, you can provide a warning.

The `CscProjectLicensing` object provides this information about the remaining volume licenses for OCR, classification and extraction, and throws events when a defined limit is reached.

To provide a license warning

1. Within an initialization function, create a new `CscProjectLicensing` object.
2. Set up the events by calling the object's method `SetLicenseWarningThreshold` and define the thresholds for all volume licenses that you want to track.

   For each processed document the event is fired when the volume license under-runs the threshold for the corresponding volume license.

3. If you do not want to issue the warning each time, but only once when starting the server, use the initialization function to check the object's `RemainingOCRLicenses` value, and output a warning to the log file.

The following visual basic code shows a sample implementation of a license warning:

```
  'Define Licenses you would like to be warn of.
    Private Enum eLicenseCodes
        eKTTPageCount = 208
        eKTT5Fields = 209
        eKTTUnlimFields = 210
    End Enum
m_ProjectLic = New CscProjectLicensing
            m_ProjectLic.ActivateLicense("", "", "", 0)
            ImageLic.ActivateLicense("", "", "", 0)
            XDocLic.ActivateLicense("", "", "", 0)
            FR7Lic.ActivateLicense("", "", "", 0)
            ' configure events for license warning
            ' when the volumes are lower than the threshold the events are fired
            'set counter when license warning should be displayed
            m_ProjectLic.SetVolumeLicenseWarningThreshold(eLicenseCodes.eKTTPageCount,
 2000)
            m_ProjectLic.SetVolumeLicenseWarningThreshold(eLicenseCodes.eKTT5Fields,
 1000)

 m_ProjectLic.SetVolumeLicenseWarningThreshold(eLicenseCodes.eKTTUnlimFields, 1000)

            ' check remaining volume
            If m_ProjectLic.RemainingLicenses(eLicenseCodes.eKTTPageCount) < 20000 Then
                Dim Msg As String
                Msg = String.Format("KTT Page Count volume license is below threshold:
 {0}", m_ProjectLic.RemainingLicenses(eLicenseCodes.eKTTPageCount))
                Log(ErrorTypes.etWarning, "Classification", Msg)
            End If
            'Do the same for each License you would like to be warn.
            If m_ProjectLic.RemainingLicenses(eLicenseCodes.eKTT5Fields) < 10000 Then
                Dim Msg As String
                Msg = String.Format("KTT 5 Fields volume license is below threshold:
 {0}", m_ProjectLic.RemainingLicenses(eLicenseCodes.eKTTPageCount))
                Log(ErrorTypes.etWarning, "Classification", Msg)
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
            Return False
        End Try
 Private Sub m_ProjectLic_LicenseVolumeWarning(ByVal VolumeID As Integer, ByVal
 RemainingExtrVolume As Integer) Handles m_ProjectLic.LicenseVolumeWarning
        Dim strDescription As String = String.Empty
        Select Case VolumeID
            Case eLicenseCodes.eKTTPageCount
                strDescription = String.Format("KTT Page Count volume license is below
 threshold: {0}", RemainingExtrVolume)
            Case eLicenseCodes.eKTT5Fields
                strDescription = String.Format("KTT 5 field volume license is below
 threshold: {0}", RemainingExtrVolume)
            Case eLicenseCodes.eKTTUnlimFields
                strDescription = String.Format("KTT unlimited field volume license is
 below threshold: {0}", RemainingExtrVolume)
            Case Else
                strDescription = String.Format("KTT volume license with ID {0} is below
 threshold: {1}", VolumeID, RemainingExtrVolume)
        End Select
```

```
        pfrmMain.Log(ErrorTypes.etWarning, "LicenseWarning", strDescription)
End Sub
```

## Warning when primary license server fails

To ensure that you have the necessary information about your license servers, you can configure notifications to warn you when the primary license server fails and switches to the backup system. An additional notification can be configured to notify you when the connection to the primary license server is restored.

The following visual basic code shows a sample implementation of a license warning notification:

```
'Event handler that warns user when connection to the
    'primary license server is lost and system automatically connects to backup
 Private Sub mLicenser_FallBackOnBackup() Handles m_ProjectLic.FallbackToBackupServer
   pfrmMain.Log(ErrorTypes.etWarning, "Licensing", "Connection to primary license
 server lost. Switching to backup license server.")
End Sub

'Event handler that warns user when connection to the
'primary license server automatically is restored
Private Sub mLicenser_Primary() Handles m_ProjectLic.BackToPrimaryServer
  pfrmMain.Log(ErrorTypes.etWarning, "Licensing", "Primary license server connection
 restored.")
End Sub
```

Chapter 8

# Access Kofax Transformation Toolkit documentation

By default, the Kofax Transformation Toolkit documentation is available online. However, if necessary, you can also configure Kofax Transformation Toolkit to use help offline.

## Default online documentation

The full documentation set for Kofax Transformation Toolkit is available from https://docshield.kofax.com/Portal/Products/KTT/7.1.0-rh4bj2m5vf/KTT.htm.

To launch the online help for the installed version of the product, click on F1 or select Help from the menu.

## Use help offline

If you do not want to use the online hosted documentation, it is possible to configure Kofax Transformation Toolkit to use help offline on your internal network.

For Kofax Transformation Toolkit, this is done in two stages.

1. Implement offline help
2. Configure Kofax Transformation Toolkit to use offline help

First, you implement how the help is accessed offline and then you configure Kofax Transformation Toolkit to use that implemented help.

To access the help offline, download it from the Kofax Fulfillment Site.

For example, download the `KofaxTransformationToolkitDocumentation_7.1.0.zip` file.

### Implement offline help

There are two ways of implementing your offline help.

1. IIS application-based offline help that is visible to your internal network only.
   Kofax recommends this method. One reason is because you do not need to copy content to multiple machines when Kofax Transformation Toolkit installed across your network.

This type of help is available in English only, except for the Thin Client Server. See the *Kofax Transformation Toolkit Thin Client Server Developer's Guide* for more information.

2. File-based offline help that is available on your internal network.
   If you cannot use the IIS application-based implementation, you can use this method to install the help on a file system visible to all Kofax Transformation Toolkit.
   This type of help is available in English only.

## IIS application-based offline help

This method of configuring your offline documentation requires that IIS is installed on your network. If you have a highly distributed Kofax Transformation Toolkit installation where components are installed on several machines, this is the easiest way of implementing offline documentation.

Kofax recommends this method for offline documentation because it mimics the behavior of the hosted site. Kofax Transformation Toolkit documentation is designed to use the hosted website, so if you require offline documentation, an internal IIS application-based solution is best. If this is not possible, you can use File-based offline help.

If you are using the Thin Client Server its documentation is installed separately. For more information see the *Kofax Transformation Toolkit Thin Client Server Developer's Guide*.

You can configure offline help to use an internal web server by following these steps:

1. If it not already installed, install the Internet Information Services (IIS) on a server on your internal network.
2. Download and extract the `KofaxTransformationToolkitDocumentation_7.1.0_EN.zip` file from the Kofax Fulfillment Site.
3. Set up the folder hierarchy for the offline documentation.
   a. Navigate to `C:/inetpub/wwwroot/` and create a folder called `Docs`.
   b. Create a second folder under `C:/inetpub/wwwroot/Docs` called `KTT`.
      The result is the following folder structure: `C:/inetpub/wwwroot/Docs/KTT`.
   c. Copy the `en_US` folder and any other language that you plan to use from the extracted zip file to `C:/inetpub/wwwroot/Docs/KTT`.
      The result of this should be one or more language folders inside the `KTT` folder.

   The folder hierarchy is set up.
4. Convert the existing folder hierarchy to an IIS application.
   a. Expand the **Default Web Site**.
      The `Docs` folder is displayed.
   b. Right-click on `Docs` and select **Convert to Application**.
      The **Add Application** window is displayed.
   c. Keep all of the default settings and click **OK**.

   The **Add Application** window closes and the `Docs` folder is converted to an IIS application.
5. Set up language redirection.

This is necessary for the KofaxTransformationToolkit even though its offline help is available in English only.

### Language redirection for IIS application-based solutions

If you are using a file system-based solution for offline documentation, the help is able to redirect automatically based on the Browser language. However, when you are using an IIS application-based solution, the help is not able to redirect you to your browser language automatically.

Even though the help is available in English only, this redirection is required to display the help.

You can configure language redirection for IIS application-based solutions by following these steps:

1. Navigate to the **redirection** folder included in the `KofaxTransformationToolkitDocumentation_7.1.0_EN.zip` file.
2. In a separate window, navigate to the application path that you created when you configured your IIS application-based offline help.
3. Move the `web.config` from the **redirection** folder to the root of your IIS application.
   For example, `C:/inetpub/wwwroot/Docs/web.config`.
4. Move the `redirection/bin` folder to the `Docs` folder to the root of your IIS application.
   For example, `C:/inetpub/wwwroot/Docs/bin/`.
5. Restart the World Wide Web Publishing service.
   Your IIS application-based offline help is configured. Note that this documentation is available in English only.

## File-based offline help

If you do not want to use an IIS application-based solution for your offline help, you can use a file system solution. For the best results, use a location that is visible across your network. If you have user modules installed at various sites, each of these need to access the location where the files are stored.

File-based offline help is supported by the design-time modules only. This means that the help is in English only.

See the *Kofax Transformation Toolkit Thin Client Server Developer's Guide* for more information about configuring offline help for the Thin Client Server.

Ensure that each client system has a default browser configured on their machine.

You can configure offline help to use a network folder by following these steps:

1. Download and extract the `KofaxTransformationToolkitDocumentation_7.1.0_EN.zip` file from the Kofax Fulfillment Site.
2. Navigate to `%ProgramData%\Kofax\Transformation\` and create a folder called `Help`.
3. Copy the `en_US` folder from the extracted zip file to the `Help` folder.
   Now that you have implemented the folder hierarchy for offline help, you need to Configure Kofax Transformation Toolkit to use offline help.

# Configure Kofax Transformation Toolkit to use offline help

In order to configure offline help for Kofax Transformation Toolkit, you need to configure the Design-time modules Help Path.

## Design-time modules Help Path

The following modules and applications fall under the design-time category.

- Project Builder
- XDoc Browser
- Image Classifier
- Statistics Viewer

> ⚠ Statistics Viewer will be deprecated and no longer available in a future version of Kofax Transformation Toolkit.

- Project Merge Tool
- OCR Server

> ℹ The design-time modules have help in English only.

You can define the Help Path for your design-time modules to use offline help by following these steps:

1. Implement offline help.
2. Open Project Builder.
3. From the **File** menu, select **Options**.
   The **Project Builder Options** window is displayed.
4. Edit the **Help Path** with one of the following.

| Setting | Description |
|---|---|
| **If you implemented your offline help to use IIS application-based offline help:** | Enter the root of the application.<br>For example, `http://<localhost>/Docs/KTT/` |
| **If you implemented your offline help to use File-based offline help** | Enter the path of the network folder where you copied the offline documentation.<br>For example, `C:\%ProgramData%\Kofax\Transformation\Help\`. |

5. **Save** your project.
6. Test the help by pressing **F1**.
   The help should open from the location configured for offline help.
   The help for these modules is available in English only. If you change the application language, the help is still displayed in English.

If an error occurs, double-check the paths to ensure that they are correct and then try again.

## Return to online hosted help

If you no longer want to use offline help, you can change your configuration and return to online hosted help by following these steps:.

For the design-time modules, do the following.

a. Open Project Builder.

b. From the **File** menu, select **Options**.
The **Project Builder Options** window is displayed.

c. Beside the **Help Path**, click **Restore Default**.
This changes the path back to its original online hosted path.

d. **Save** your project.

The help accessed by all of the design-time modules is now the online hosted help.