

# Nuance OmniPage Capture SDK 20

## What's New

This document presents an overview of the innovations, improvements and changes in version 20 of the OmniPage Capture SDK (OP SDK).

The online documentation for the chosen platform provides full information on these components. New features of earlier versions can be found in the documents of the given release.

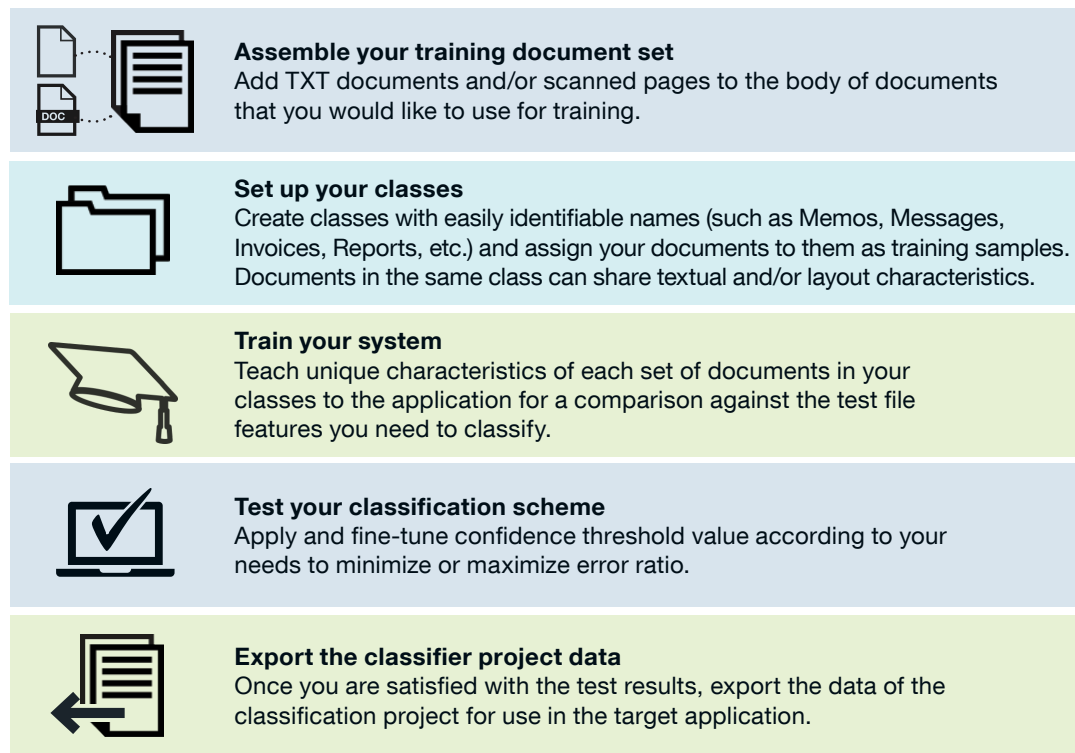
## 1. Document Classifier Assistant Concept

### 1.1 In a nutshell

Documents exist in many different types and layouts and it is a common task to identify and/or categorize them based on their diverse contents. Use Document Classifier's supervised learning and pattern recognition model and tools to set up a project for such tasks. Some common use-case examples include spam filtering, automatic e-mail forwarding, mail-room automation and big data search.

### 1.2 How it works

The outline of the Document Classifier workflow is the following:



### 1.3 Overview

#### What Document Classifier is NOT

It is **NOT** an application for rule-based classification. In other words, you do not have to painstakingly create rules specifying, for instance, a given piece of text to be located at special coordinates on the page in order to classify this page as a member of one of your classes. No unsupervised learning, no automatic clustering.

#### What Document Classifier IS

It **IS** an application where you simply assign documents to classes, then train the system on class characteristics, and finally you test and fine-tune your scheme.

Visual user interface for a set of API calls.

## 2. Intelligent Workflow Runner

### 2.1 Introducing Intelligent Workflow Runner component of the OP SDK V20

Intelligent Workflow Runner (component name is OCRService) is a high availability API for long running OP SDK applications with scale-up capability and declarative programming method.

Workflow Runner is implemented as an out-proc COM object. It can be used in a standard way from .NET or native/Win32 or any other programming environment that can work with COM objects. Workflow Runner has a very simple API that only dispatches the jobs to be processed and provides a programming space. The actual OP SDK workflow is given in an XML string in a declarative way.

#### 2.1.1 Job items

The XML string consists of several predefined job items, one being the dedicated launcher for all the others. Each job item's successful operation depends on the successful completion of the previous job item. They can query the result of their dependent previous job item's status in a response XML.

If the Workflow Runner encounters any errors, it switches off the OCRServer responsible for the job processing. The following error types can be manifested:

- Start error: job execution is not started
- Start errors (array): error of a job item
- Runtime error: signal minor errors that do not influence successful job completion (e.g. no text on a page)

Workflow Runner has many built-in commonly used OP SDK workflows, like converting images to DOCX or searchable PDF. These are basically IProPlus workflows. A WorkflowXML string describes workflows and their parameters. It also names the input-output files and sets OP SDK settings. After a WorkflowXML has been processed, a response XML string returns to the application with statistics and errors, if any.

There are two ways to create WorkflowXML files. The AssistantApp.exe should be used to create classic workflows (binary xwf file) and a tool XWFToXML.exe to convert .xwf files to WorkflowXML files. The second way is to use the Workflow XML Designer application. It is a Windows Workflow Foundation program. You can create workflows using the application's UI and save them as WorkflowXML files. The content of a WorkflowXML file can be given a first parameter of Run() function.

The Workflow Runner (OCRService.exe) process maintains multiple OP SDK worker processes (OCRServer.exe). It stops and restarts workers on a regular basis (default: every 10 WorkflowXML execution) and at every

occurrence of an OP SDK error, crash, hang or infinite loop. Before a restart, Workflow Runner cleans the TEMP folder of the actual worker. This mechanism ensures the high availability and the safety of the application. No resource leaks, hangs, or crashes are manifested.

The Run (string ID, string WorkflowXML) method of the Workflow Runner API is an asynchronous process. There is a done event to notify the application about the end of a WorkflowXML. Workflow Runner maintains a small queue of WorkflowXMLs feeding the OP SDK worker processes. The number of workers can be parameterized. Each OP SDK process utilizes one to five CPU cores depending of the workflow steps.

If the computer node has many cores, the application can increase (UserCount) the number of the worker processes (scale-up). In general, the application can set one worker for every two available virtual cores. If the node has to deal with another task, the application can limit the workers to one or two. Using a parameter in the WorkflowXML, the application can reduce (throttle) the CPU usage of a single worker if needed.

### 2.2 Workflow Runner usage

- The two out-proc COM processes must be registered (at admin prompt or in the application installer):
  - OCRService/regserver
  - OCRServer/regserver
 (Note: by default the OP SDK Installer does not register these)
- In VisualStudio, add Nuance OmniPage CSDK OCRService 20.0 Type Library to references of your application. (In COM tab of the **Reference Manager** dialog)
  - From a C++ program, use `#import "OCRService.exe" named_guides` to access OCRService COM.
- Set OCRServiceLib Embedded Interop Types property to False.
- Implement the application.

Use the provided **Workflow XML Designer** tool with the included sample workflow creating XML files as a guideline.

### 2.3 Advanced usage of Workflow Runner

OCRService.exe can be hosted in a Windows NT Service (maintaining OCRService process life cycle). You can easily scale out Workflow Runner by using multiple computers storing and distributing WorkflowXMLs (and response XMLs) through durable data store components like databases, queues, ServiceBus or Azure ServiceBus. In such a system there can be multiple nodes (scale-out) each running with an OCRService.exe instance with multiple workers.

## 2.4 Intelligent Workflow Runner API

```
void SetLicense(string LicenseFile, string Key)
void Run(string XML, string ID)
void Cancel(string ID)
void Ping(bool Report)
delegate void IOCRServiceEvents_
DoneEventHandler(string ID, int StartError,
System.Array StartErrors, System.
Array RuntimeErrors, System.Array
RuntimeErrorDescriptions, System.Array
ResponseXMLFiles)

int TimeOut { get; set; }//maximum JobXML
execution time
int UserCount { get; set; }//worker count
int RpcTimeOut { get; set; }//out-proc COM API
time out
int PendingUsers { get; }//queue length
int Lives { get; set; }//JobXMLs between two
restarts
bool Log { get; set; }//turn the log off
```

## 3. Other major changes

- This release the OP SDK introduces Server and Desktop type licensing. On server type environments Nuance Central Licensing Service (NCLS) can be used. It is very similar to Nuance Licensing Service and Nuance Licensing Tool.

In addition, from v20 OP SDK starts to count processed pages. OP SDK Customers can have Page Pack and Page Per Month licenses. Both have final expiration dates. In server type environment the NCLS accepts license requests from multiple OP SDK processes running on other computers in the LAN. NCLT creates monthly usage reports to help OP SDK customers to follow the processed page consumption. NCLT can be set up to send notification emails about reaching page limits.

Licensing desktop type application (runtime) the situation remains very similar. Desktop licensing also counts the processed pages and prepares monthly reports. This type of licensing does not require .NET Framework.

- Enhanced performance: this release includes general stability and functionality bug fixes
- Asian zoning improvements
- JPEG2000 Engine updated, Kakadu version updated to 7.8
- Nuance Vocalizer 2.0/2.1 support for premium quality audio – available for 32-bit and 64-bit versions. To access Vocalizer files, you need to install and license the Vocalizer engine – this is not part of this OP SDK release.
- Logical Form Recognition Engine can now recognize rounded check boxes, or radio buttons.
- DOCX, XLSX, PPTX do not require .NET Framework installed
- RER Handwritten OCR engine (ICR2) updated. (This engine has been built using VS2013. If your distribution uses the RER engine, your installer has to install the vc\_redist of VS2013.)
- Project files for samples are now in VS2010 format. This makes it easy to update these files to the version of Visual Studio you are using. Samples binaries were also built using VS2010. That is why OP SDK Installer installs vc\_redist for VS2010.
- ReclProPlus API retired: the ReclProPlus component has been removed from this release. If you still intend to use it, the current OP SDK release contains the obsolete component source files. Locate these (ReclproPlus.cpp and ReclproPlus.h) at the following path: **<CSDK Installation Path>\Samples\ISApp\MFC.**
- AmpLib Barcode Recognition Engine is discontinued BAR\_C32, BAR\_4STATE, BAR\_4STATE\_DK1, BAR\_AZTEC bar types are no longer supported.

---

### About Nuance Communications, Inc.

Nuance Communications, Inc. is a leading provider of voice and language solutions for businesses and consumers around the world. Its technologies, applications and services make the user experience more compelling by transforming the way people interact with devices and systems. Every day, millions of users and thousands of businesses experience Nuance's proven applications. For more information, please visit: [www.nuance.com](http://www.nuance.com).

---